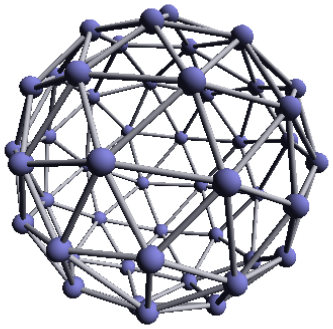
Problema de Thomson



Renato Paes Leme

18 de dezembro de 2007

Sumário

0.1 Introdução

J. J. Thomson propôs em 1897 o modelo do "pudim de Natal" para o átomo. Nesse modelo, os elétrons eram "corpúsculos" flutuando em uma sopa com carga elétrica positiva uniformemente distribuída. Na analogia com o "pudim", os elétrons eram como as passas e a sopa positiva a massa do pudim. Um problema interessante nesse modelo é calcular as configurações de menor energia (*ground state*) de como os elétrons podem se distribuir no "pudim". Considerando o modelo de força elétrica de Coulomb, temos que duas partículas de mesma carga se repelem com força:

$$F = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r^2}$$

onde r é a distância entre as partículas. A força de Coulomb é um campo gradiente, ou seja, fixada uma partícula, podemos escrever $F = \nabla U$, assim temos uma função de energia potencial. Para o conjunto de partículas, temos:

$$U(x^1, ..., x^n) = \sum_{i<j} \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\|x^i - x^j\|}$$

onde $x^i \in \mathbb{R}^3$. Considerando que as partículas estão confinadas a uma bola $B_r := \{x : \|x\| \leq r\}$, e sendo a força de repulsão, nossa intuição nos diz que as partículas tendem à fronteira. Assim, podemos simplificar o problema, supondo

inicialmente as partículas já na fronteira (e, de fato, estando lá, como a força age na reta que liga as partículas e B_r é convexo, a força faz com que as partículas permaneçam na fronteira). Assim, podemos considerar o problema de minimizar a função de energia na esfera $\mathbb{S}^2$. Considerando todas as partículas de mesma carga (como no modelo de Thomson) podemos retirar as constantes e considerar o problema:

$$\min \sum_{i<j} \frac{1}{\|x^i - x^j\|} \text{ s.a. } \|x^i\|^2 = 1 \tag{1}$$

onde $x^i \in \mathbb{R}^3$. Assim, temos um problema de otimização com restrições de igualdade em $\mathbb{R}^{3n}$ onde n é o número de partículas.

Mesmo não tendo sido o modelo de Thomson muito bem sucedido no objetivo de explicar as propriedades do átomo (logo Rutherford mostrou, com sua famosa experiência, que a maior parte do átomo é vazia), o problema de Thomson ainda tem aplicações importantes em biologia computacional e em química, por exemplo, no arranjo de proteínas que formam a membrana de vírus esféricos ou na teoria VSEPR (Valence shell electron pair repulsion).

Outra propriedade interessante é que este problema tem uma natureza combinatória muito interessante. Intuitivamente, as soluções globais de (1) nos dão uma forma de distribuir de forma homogênea partículas sobre uma esfera. Dessa forma, esperamos que para uma quantidade suficiente de pontos, o fecho convexo de $\{x^1, ..., x^n\}$ seja uma boa aproximação discreta para a esfera. Não é, portanto, surpreendente que as soluções globais do problema de Thomson para $n = 4, 8, 12, 20$ sejam os poliedros de Platão correspondentes.

Para resolver esse problema, utilizaremos o algorimo ALGENCAN. Discutiremos brevemente esse algoritmo e apresentaremos em seguida os resultados obtidos aplicando este método para o problema (1) tal como enunciado anteriormente e com ligeiras modificações. Foi utilizada a implementação do ALGENCAN disponível na biblioteca TANGO - Trustable Algorithms for Nonlinear General Optimization [1], originalmente escrita em Fortran. Neste trabalho foi usada a interface C para a biblioteca. Para gerar a visualização, usou-se OpenGL.

0.2 Algoritmo ALGENCAN

0.2.1 Overview do Algoritmo

O ALGENCAN ([ABMS07] e [ABMS05]) é um algoritmo desenvolvido para resolver problemas de otimização com restrições de igualdade e desigualdade, seguindo a filosofia dos métodos de Lagrangeano Aumentado. No nosso caso, temos apenas restrições de igualdade, portanto vamos discutir esse caso mais específico, ou seja, do problema:

$$\min f(x) \text{ s.a. } h(x) = 0 \tag{2}$$

onde $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ e $h : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ são funções de classe $\mathcal{C}^1$. Esse problema é resolvido, minimizando o Langrangeano aumentado:

[1] Disponível em *www.ime.usp.br/∼egbirgin/tango/*

$$L_\rho(x,\lambda) := f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \frac{\rho}{2}\|h(x)\|^2 \tag{3}$$

A essência desse método é usar o último termo penalizar a violação de viabilidade. Quando $\rho \to \infty$, os mínimos de $L_\rho(\cdot,\lambda)$ tendem a pontos viáveis (pois o termo de penalização torna o valor de L_ρ nos pontos não viáveis muito grande). Mais formalmente, se $\rho_k \to \infty$ e $\{\lambda_k\}$ é um conjunto limitado e:

$$x^k \in \text{argmin}_x L_{\rho_k}(x,\lambda^k)$$

então todo ponto de acumulação de $\{x^k\}$ converge a uma solução global de (2). A prova desse fato é bastante simples e pode ser encontrada em [Ber99]. Sob as mesmas condições, se tivermos $x^k \to x^*$ temos:

$$\lambda^k + \rho_k h(x^k) \longrightarrow \lambda^*$$

Assim, os métodos de Langrangeano aumentado permitem que recuperemos tanto a solução ótima quanto os multiplicadores de Lagrange.

O ALGENCAN usa uma formulação ligeiramente diferente do Lagrangeano Aumentado do que (3). Dada por:

$$L(x,\lambda,\rho) := f(x) + \frac{\rho}{2}\sum_{i=1}^{m}\left(h_i(x) + \frac{\lambda}{\rho}\right)^2 \tag{4}$$

Um método típico de Lagrangeado Aumentado é composto de Iterações Exteriores e Interiores. Começamos com um par (x^0,λ^0). Em uma Iteração Exterior é achado um mínimo global (ou um bom mínimo local) de $L_\rho(\cdot,\lambda^k)$ e em seguida atualizamos os multiplicadores e os parâmetros de penalidade. É esperado que as Iterações Externas proporcionem uma melhora de *complementariedade* e/ou *viabilidade*, ou seja, que os termos $\lambda^\top h(x)$ e $\|h(x)\|$ sejam menores.

Note que em muitos casos, é interessante tomar parâmetros de penalização diferentes para cada restrição. Assim, em (4) substituimos ρ por ρ_i.

De forma geral, são usadas regras clássicas de correção de primeira ordem para os multiplicadores de Lagrange (como indicado em [ABMS07]) e é imposta a condição de que os multiplicadores devem pemanecer limitados.

As iterações Interiores correspondem à minimização (irrestrita ou em um conjunto simples, como uma caixa) do Langrangeano Aumentado. O ALGENCAN usa o GENCAN nessa etapa, um método de minimização em uma caixa que utiliza o Método do Gradiente Espectral Projetado (ver [BM02]) para mais detalhes.

0.2.2 Principais Parâmetros

O ALGENCAN recebe os seguintes parâmetros:

- ponto inicial $x^0 \in \mathbb{R}^n$
- $\tau \in [0,1)$
- $\gamma > 1$
- $\rho > 0$

- bounds $\lambda_i^{\min} < \lambda_i^{\max}$
- $\lambda^0 \in \mathbb{R}^m$ tal que $\lambda_i^{\min} < \lambda_i^0 < \lambda_i^{\max}$
- sequência $\{\varepsilon_k\}$ com $\varepsilon_k \to 0$

O algoritmo inicializa x e λ com os valores iniciais dados. Em seguida, para cada iteração, o algoritmo tenta achar x^k :

$$\|\nabla L(x^k, \lambda^k, \rho_k)\| \leq \varepsilon_k \tag{5}$$

Em seguida atualizamos os multiplicadores, fazendo λ^{k+1} igual a $\lambda^k + \rho_k h(x^k)$ e projetando cada componente no intervalo $[\lambda_i^{\min}, \lambda_i^{\max}]$. A atualização dos parâmetros de penalização se dá da seguinte forma: se $\|h(x^k)\|_\infty \leq \tau \|h(x^{k-1})\|_\infty$ então $\rho_{k+1} = \rho_k$. Senão, fazemos $\rho_{k+1} = \gamma \rho_k$

Ou seja, verificamos se houve ganho significativo de viabilidade - se houve, mantemos o parametro de penalização, senão, aumentamos ele de um fator γ. No código, chamamos τ de RHOFRAC e γ de RHOMULT.

0.3 Solução Direta

0.3.1 Definindo as funções

Para resolver o problema de Thomson (1) usando ALGENCAN, temos que primeiro definir a função objetivo e a função que define as restrições, bem como o gradiente e hessiana destas. Uma opção que o software desenvolvido por Martinez e Birgin nos dá é a de usar diferenças finitas para calcular o gradiente e a hessiana. Como no caso podemos fornecer esses valores exatamente, temos maior precisão e menor custo computacional do que se tivéssemos que calculá-los por diferenças finitas. Temos:

$$f(x^1, ..., x^n) = \sum_{i<j} \frac{1}{\|x^i - x^j\|}$$

$$\nabla_{x^i} f(x^1, ..., x^n) = \sum_{i \neq j} \frac{x^j - x^i}{\|x^i - x^j\|^3}$$

$$\nabla^2_{x^i x^i} f(x^1, ..., x^n) = \sum_{i \neq j} -\frac{I}{\|x^i - x^j\|^3} + 3\frac{(x^i - x^j)(x^i - x^j)^\top}{\|x^i - x^j\|^5}$$

$$\nabla^2_{x^i x^j} f(x^1, ..., x^n) = \frac{I}{\|x^i - x^j\|^3} - 3\frac{(x^i - x^j)(x^i - x^j)^\top}{\|x^i - x^j\|^5}, \quad i \neq j$$

As restrições são dadas por $c : \mathbb{R}^{3n} \longrightarrow \mathbb{R}^n$, onde a k-ésima restrição diz que a k-ésima partícula está na esfera $\mathbb{S}^2$.

$$c_k(x^1, ..., x^n) = \|x^k\|^2 - 1$$

$$\nabla_{x^i} c_k(x^1, ..., x^n) = \begin{cases} 2x^i & i = k \\ 0 & i \neq k \end{cases}$$

$$\nabla^2_{x^i x^j} c_k(x^1, ..., x^n) = \begin{cases} 2I & i = j = k \\ 0 & \text{caso contrário} \end{cases}$$

0.3.2 Experimentos Iniciais

Foi feito um primeiro teste com o ALGENCAN definindo, além da função, o gradiente e a hessiana (para não precisarmos usar diferenças finitas). Usamos a opção HPTYPE = 4, que significa que aproximamos a Hessiana por uma correção BFGS de uma aproximação Gauss-Newton da Hessiana. Usa-se também uma correção espectral para forçar que a aproximação Gauss-Newton seja definida positiva. Foi usado nesse primeiro experimento RHOMULT = 10 e RHOFRAC = 0.5, deixando o algoritmo escolher automaticamente o parâmetro de penalização. Daqui para a frente, quando nada for dito, estamos usando os parametros descritos acima no ALGENCAN para o problema (1). Só serão mencionadas as características de cada execução que diferem dos parâmetros apresentados acima.

Os pontos iniciais são gerados aleatoriamente sobre a esfera $\mathbb{S}^2$. Intuitivamente, esperamos que a solução do problema (1) seja um conjunto de pontos bem distribuídos na esfera, logo é bem natural já começarmos com um conjunto de pontos bem espalhados. Utilizamos o seguinte algoritmo para gerar um ponto $p \in \mathbb{S}^2$ de tal forma que sua distribuição de probabilidade seja uniforme na esfera: Consideramos um mapa $g : [0, 2\pi] \times [0, \pi] \longrightarrow \mathbb{S}^2$ dado por:

$$g(\theta, \phi) = (\cos\theta \sin\phi, \sin\theta \sin\phi, \cos\phi) \tag{6}$$

O elemento de volume nessa superfície é dado por: $\|g_\theta \times g_\phi\| = |\sin\phi|$, logo a distribuição de probabilidade de (θ, ϕ) deve ser proporcionar a $|\sin\phi|$. Uma maneira de fazer isso, tendo uma função `Uniforme()` que retorna um valor uniformemente distribuído no intervalo $[0, 1]$ e uma função `Moeda()` que retorna $k \in \{\texttt{verdadeiro}, \texttt{falso}\}$ com igual probabilidade, podemos fazer da seguinte forma:

$\phi = arccos($ `Uniforme()` $)$
Se (`Moeda()`) então $\phi = \pi - \phi$
$\theta = 2\pi$ `Uniforme()`

Esse procedimento leva a uma distribuição uniforma na superfície da esfera, pois, analisando $\phi \in [0, \pi/2]$, temos:

$$P(\phi < \overline{\phi}) = \frac{1}{4\pi} \int_0^{\overline{\phi}} \int_0^{2\pi} \sin\phi d\theta d\phi = \frac{1}{2}(1 - \cos\overline{\phi})$$

Assim, podemos ver que o algoritmo apresentado produz exatamente a distribuição de probabilidade em ϕ correspondente à distribuição uniforme na superfície da esfera. Os resultados dessa abordagem se encontram na Tabela 1, onde n é o número de partículas, f^* é a solução ótima encontrada pelo ALGENCAN, OI é o número de Iterações Externas (Outer Iterations), II é o número de Iterações Internas (Inner Iterations), LFE é o número de vezes que o Lagrangeano foi avaliada, LGE o número de vezes que o gradiente do Lagrangeano foi avaliado e tempo é o tempo de execução do algoritmo em segundos.

Este experimento, bem como os demais, tiveram seu tempo medido em um AMD 64bits com 1GB de memória RAM. Os resultados se encontram documentados na Tabela 1. Uma checagem dos resultados pode ser feita em [HSS97]. Foram conseguidos nesse experimento alguns resultados ligeiramente melhores que os documentados em [HSS97].

Por esta tabela, o valor ótimo apresenta um bom comportamento com n. O comportamento se assemelha muito a uma quadrática, ao menos para n entre 3 e 100. O ajuste de um polinômio quadrático aos dados da tabela acima nos dá o polinômio: $0.4692n^2 - 2.6093n + 12.923$ com coeficiente de determinação $R^2 \approx 1$. O gráfico do valor de f^* como função de n se encontra na Figura 1.

Tabela 1 - Primeiros Experimentos

n	f^*	OI	II	LFE	LGE	tempo
3	1.732	3	14	39	22	0.01
4	3.674	4	22	40	34	0.01
5	6.474	4	25	38	34	0.01
6	9.984	4	23	57	35	0.01
7	14.452	5	55	141	72	0.01
8	19.674	5	47	91	60	0.01
9	25.759	6	53	96	70	0.02
10	32.716	6	57	125	77	0.02
11	40.595	6	75	126	94	0.03
12	49.163	6	50	109	67	0.03
13	58.8522	7	99	193	120	0.03
14	69.3044	7	63	104	83	0.03
15	80.6668	7	86	148	108	0.05
16	92.9186	8	77	117	96	0.05
17	106.047	8	126	279	150	0.08
18	120.080	8	99	201	124	0.05
19	135.087	9	193	454	222	0.13
20	150.878	9	93	161	117	0.06
21	167.636	9	191	417	221	0.15
22	185.279	9	105	177	128	0.08
23	203.925	10	110	204	136	0.09
24	223.34	10	85	147	111	0.06
25	243.802	10	141	286	167	0.13
26	265.127	11	129	238	161	0.10
27	287.294	11	101	166	129	0.10
28	310.479	11	102	189	134	0.10
29	334.618	11	213	436	244	0.22
30	359.593	12	164	326	202	0.16
31	385.517	12	100	181	131	0.13
32	412.243	12	96	176	128	0.09
33	440.195	11	187	384	219	0.25
34	468.889	8	160	315	187	0.19
35	498.566	8	246	530	280	0.39
36	529.118	8	303	643	331	0.45
37	560.622	8	210	424	237	0.33
38	593.026	7	139	254	161	0.17
39	626.374	7	121	193	139	0.19
40	660.657	7	136	220	154	0.21
41	695.895	7	127	212	148	0.20
42	732.053	7	104	176	125	0.16
43	769.161	7	370	818	402	0.69

44	807.139	7	267	566	292	0.51
45	846.148	7	367	808	398	0.80
46	886.167	7	170	311	188	0.41
47	927.050	7	293	642	318	0.67
48	968.700	7	151	247	173	0.29
49	1011.54	7	133	235	153	0.35
50	1055.16	7	145	251	165	0.36
51	1099.80	7	167	310	188	0.43
52	1145.41	7	206	372	230	0.50
53	1191.9	7	209	403	227	0.55
54	1239.34	7	244	484	269	0.62
55	1287.74	7	183	342	202	0.47
56	1337.06	7	193	366	212	0.60
57	1387.34	7	166	323	187	0.43
58	1438.59	7	204	410	233	0.73
59	1490.72	7	209	433	232	0.60
60	1543.77	7	207	425	228	0.64
61	1597.87	7	236	535	261	0.77
62	1652.85	7	343	777	371	1.06
63	1708.86	8	269	566	296	0.93
64	1765.80	8	198	419	221	0.64
65	1823.65	8	396	1012	424	1.36
66	1882.42	8	442	1047	475	1.72
67	1942.10	8	156	341	178	0.52
68	2002.85	8	194	560	218	0.75
69	2064.51	8	212	612	237	0.81
70	2127.07	8	274	695	302	1.22
71	2190.61	8	235	794	261	0.87
72	2255.22	8	367	1208	401	1.40
73	2320.59	8	632	3173	690	2.55
74	2387.04	8	454	2198	497	1.81
75	2454.31	8	595	3444	654	2.58
76	2522.61	8	1766	6210	1846	6.81
77	2591.84	9	467	2148	514	1.91
78	2662.03	9	692	3058	748	3.21
79	2733.23	9	1131	3888	1194	5.19
80	2805.42	9	1074	3021	1133	5.37
81	2878.56	9	1797	5753	1891	8.34
82	2952.55	9	1148	3100	1217	5.64
83	3027.50	9	1381	4145	1450	7.24
84	3103.45	9	948	2792	1003	4.91
85	3180.33	9	1873	6336	1985	9.74
86	3258.19	9	1261	3945	1329	7.24
87	3336.96	9	1364	4393	1442	7.71
88	3416.73	9	1078	3167	1129	6.5
89	3497.39	9	1842	7007	1940	9.27
90	3579.06	8	1664	5887	1757	9.06
91	3661.84	8	2230	6841	2343	16.15
92	3745.22	8	850	2995	893	5.06
93	3829.82	9	1134	3928	1201	6.89

94	3915.33	9	790	2729	829	4.68
95	4001.78	9	1247	3868	1310	6.91
8 96	4089.13	9	1146	3368	1216	7.32
97	4177.51	9	1353	3950	1413	8.94
98	4266.79	9	1202	3562	1271	7.78
99	4357.11	9	1843	5575	1929	12.64
100	4448.40	10	1748	5376	1840	12.08

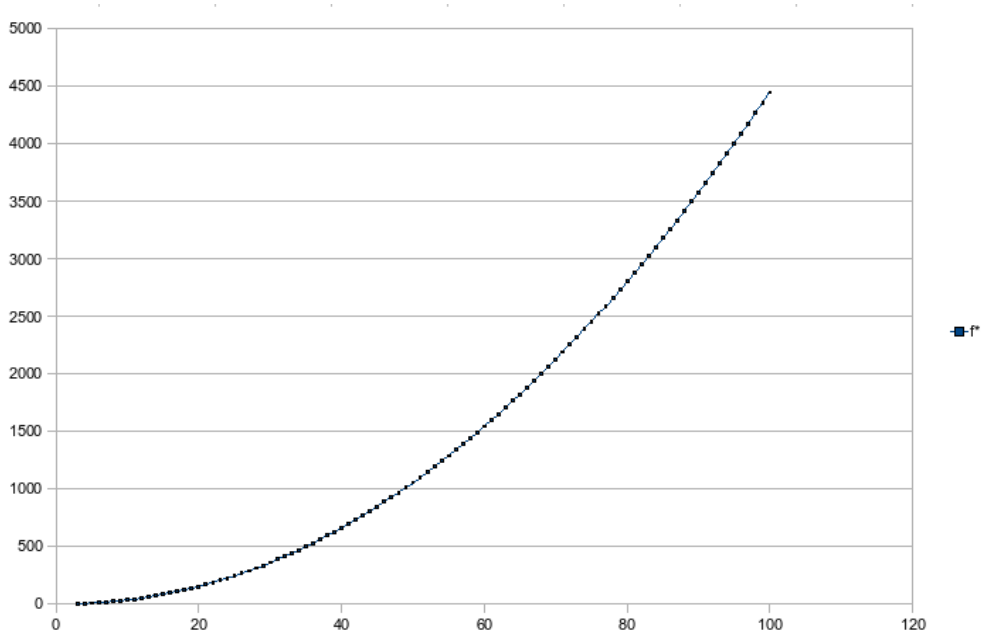


Figura 1: Comportamento de $f^*(n)$

A análise dos demais dados feita na Figura 2 (o tempo se encontra no eixo-y secundário) nos mostra que eles são altamente correlacionados - de fato, são todos medidas do esforço computacional para a execução do algoritmo. Isso é intuitivo - afinal, o número de vezes em que o Lagrangeano (ou seu gradiente) é avaliado, guarda uma relação íntima com o número de iterações internas. E quanto mais iterações do algoritmo, maior o tempo de execução. De todas essas medidas, o tempo é menos confiável, pois não é independente da plataforma de execução e sua medição é incerta, pois é difícil separar o tempo em que o algoritmo está de fato executando de outras tarefas do processador.

É curioso que não há um padrão claro nas medidas de esforço computacional. Globalmente, vemos um aumento do esforço computacional com o crescimento do n, mas mesmo assim, valores muito próximos de n têm medidas de esforço associadas bastante distintas. Por exemplo, para $n = 19$ o tempo de execução é 0.13 segundos, para $n = 20$ temos 0.06 segundos e para $n = 21$ um valor novamente alto de 0.15 segundos. Como os pontos iniciais são escolhidos aleatoriamente, podemos creditar essas diferenças aos valores iniciais. Poderíamos ter escolhido, por exemplo, um conjunto inicial de posições bons para as partículas

no caso $n = 20$ e conjuntos não tão bons para $n = 19$ e 21. Vamos repetir os experimento para alguns valores (entre 15 e 25) várias vezes e tomarmos a média. Na Tabela 2, executamos o algoritmo 50 vezes para cada caso, com um conjunto de pontos aleatórios diferente em cada caso. Os resultados estão plotados na Figura 3.

A média e o desvio padrão em 50 execuções nos sugerem que realmente o problema com $n = 20$ é mais fácil que os prolblemas com $n = 19$ e 21. A razão para isso não é clara. Me indago se pode haver alguma relação com o fato de existir um poliedro de Platão com 20 vértices.

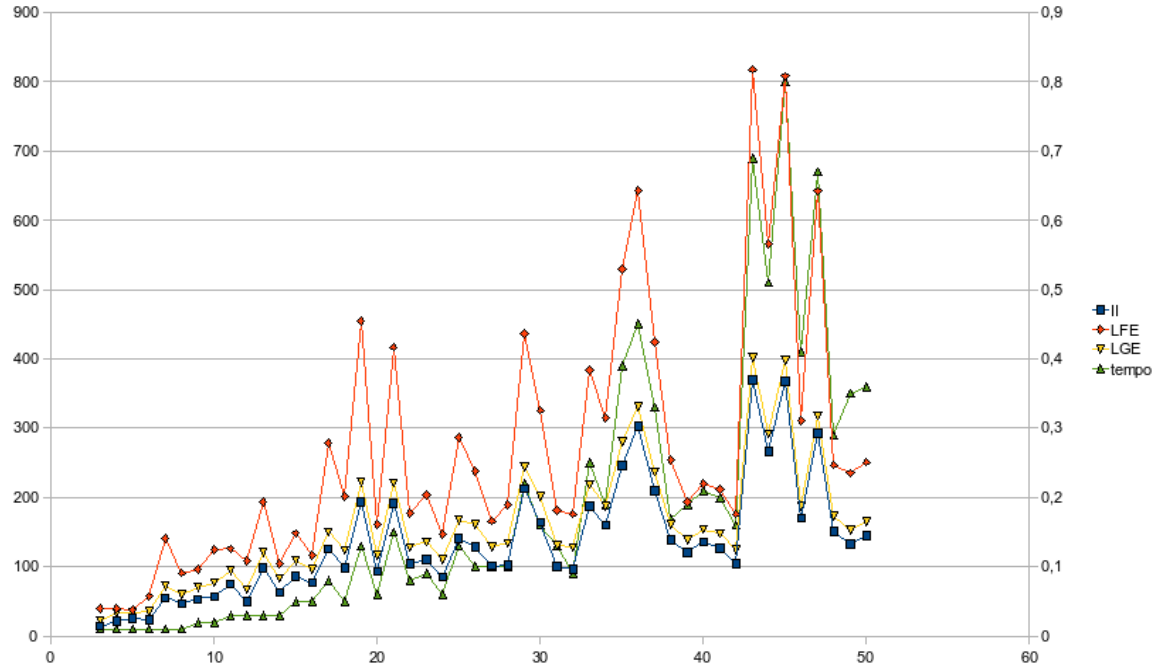


Figura 2: Medidas do tempo de execução do ALGENCAN

Tabela 2 - Média de 50 execuções

n	LFE medio	LFE desvio padrão
15	142.02	26.6169
16	137.26	22.0752
17	169.7	30.0701
18	160.96	29.2711
19	393.7	58.7518
20	182.68	25.5001
21	344	78.9159
22	194.62	40.1517
23	186.98	31.0776
24	170.58	18.5947
25	429.6	119.056

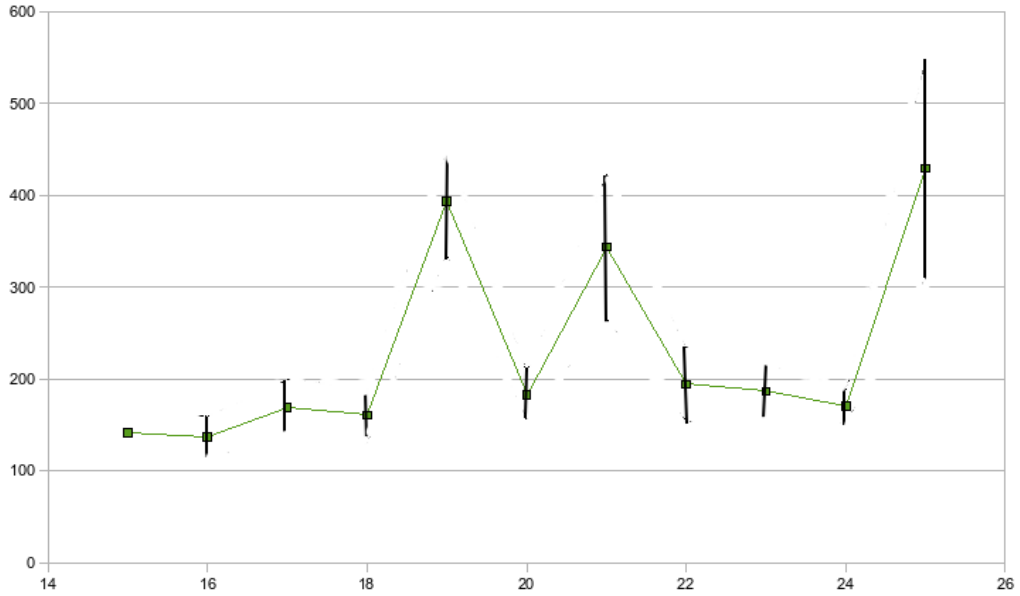


Figura 3: Media e desvio padrão de LFE em 50 execuções

0.3.3 Escolhendo os pontos iniciais

Alguns experimentos foram feitos estudando-se outras formas de se escolher os pontos iniciais. O primeiro deles foi o seguinte: utilizar valores da solução para n partículas na execução do algoritmo para $n+1$ partículas. Se $(x^1, ..., x^n)$ é a solução do problemas com n partículas e $\lambda \in \mathbb{R}^n$ é o conjunto de multiplicadores de Lagrange, fornecemos o $(x^1, ..., x^n, y)$ e $(\lambda, 0)$ como entrada da iteração seguinte (onde $y \in \mathbb{S}^2$) é um ponto tomado aleatoriamente na esfera. Os resultados desse experimento indicaram um comportamento semelhante ao anterior, ou seja, não há melhora significativa na maioria dos casos. Uma comparação das LFE com os pontos tomados aleatoriamente (linha azul) e usando o resultado do algoritmo para o n anterior (linha vermelha) se encontra na Figura 4.

Outro experimento interessante é analisar o comportamento do método próximo a pontos estacionários (principalmente próximo a mínimos locais, que não são o mínimo global). Considere as seguintes duas configurações estacionárias:

- Arranjo dos pontos no equador: Os n pontos são dados por $x^k = (\cos\theta_k, \sin\theta_k, 0)$, onde $\theta_k = 2\pi k/n$. Esse é um mínimo local do problema : se fato, é o mínimo global do problema para $n = 3$ e o mínimo global do problema restrito a $\mathbb{S}^1$ (Figura 5.d).
- $n-2$ pontos distribuidos uniformemente no equador ($\mathbb{S}^2 \cap \{(x, y, z) : z = 0\}$) e os dois pontos restantes nos polos, i.e., $(0, 0, 1)$ e $(0, 0, -1)$ (Figura 5.c). Essa configuração é o mínimo global do problema para $n = 5$ (di-piramide triangular) , 6 (octaedro) , 7 (di-piramide pentagonal) .

Começando nessa configuração, o método converge sempre para esses mínimos locais (isso foi testado para $n < 100$). O próximo passo é tomar os pontos

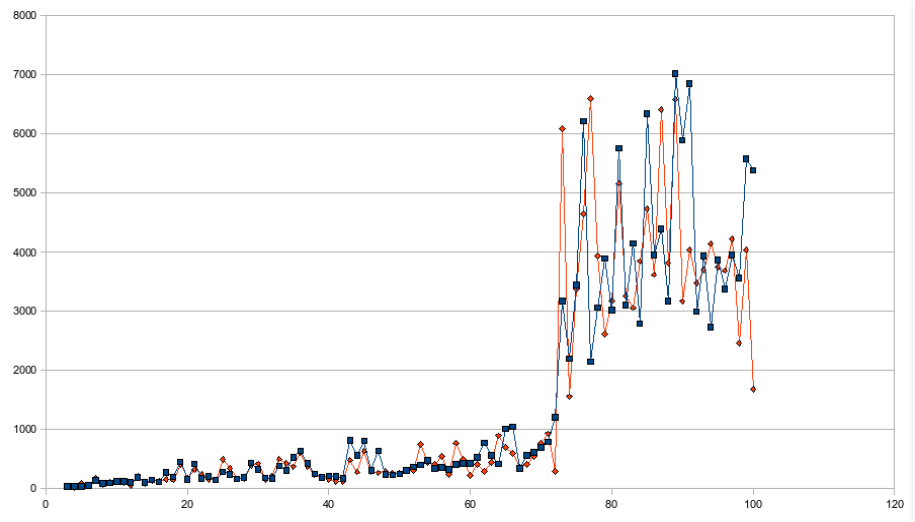


Figura 4: Dois modos de se escolher os pontos iniciais

iniciais não nessas configurações, mas em uma pequena perturbação dessas configurações, e ver se eles de fato voltam a elas ou se eles tendem ao mínimo global. Tomamos:

$$x^k = (\cos(\theta_k + \delta\theta)\sin(\pi/2 + \delta\phi), \sin(\theta_k + \delta\theta)\sin(\pi/2 + \delta\phi), \cos(\pi/2 + \delta\phi))$$

onde $|\delta\theta|$ e $|\delta\phi|$ são ângulos muito pequenos. Nesses experimentos se constatou que para perturbações pequenas ($|\delta\phi|, |\delta\theta| \approx 10^{-4}$) o caso $n = 4$ continua convergindo para o minimo local com todas as particulas no equador - aumentanto ligeiramente a perturbação, o método passa a convergir para a solução global - o tetraedro. No caso $n = 5$ é necessária uma perturbação ainda menor para que o método permaneça preso na solução local. Para $n = 5$, perturbações de $|\delta\phi|, |\delta\theta| \approx 10^{-5}$ fazem com que o método convirja para a solução local. Para $n = 6$, uma perturbação da ordem de 10^{-5} não é mais suficiente.

Adicionamos uma perturbação semelhante no arranjo com $n-2$ elétrons no equador e 2 nos pólos. Para $n \leq 8$ uma perturabação da ordem de 10^{-4} faz com que o método convirja para a solução local. Para $n \leq 7$ isso é de certa forma esperado, pois esta é a solução global. Para $n = 8$, a solução global é o anti-prisma quadrado, mas, mesmo assim, para perturbações pequenas, conseguimos convergência para esse mínimo local. Para $n > 8$, o método converge para a solução global. Ou seja, a bacia de atração dessa solução é maior que a da solução com todos os pontos no equador. Isso é, já de certa forma, esperado. Fisicamente, essa configuração é mais estável que a configuração com todos os pontos no equador. A situação física de n cargas distribuídas uniformemente no equador gera um equilíbrio delicado - qualquer pequeno deslocamento tira as cargas do equilíbrio. Assim, é esperado que a bacia de atração dessa solução não seja muito grande.

A Tabela 3 ilustra o valor ótimo obtido e o tempo de processamento para o ALGENCAN mudando apenas o modo como os valores iniciais são escolhidos.

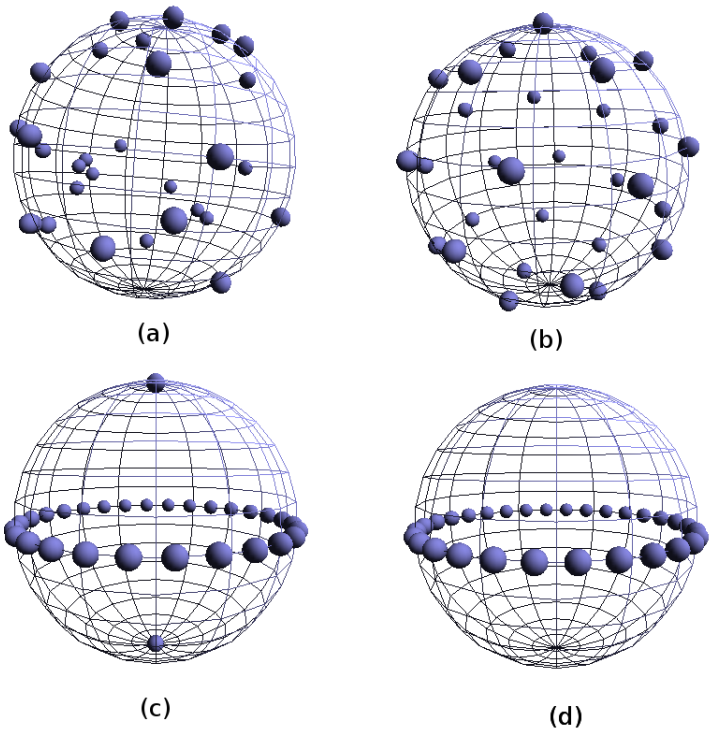


Figura 5: (a) Configuração inicial com os pontos sorteados aleatoriamente; (b) solução para o problema com pontos iniciais dados por (a); (c) arranjo com $n-2$ pontos no equador e 2 pontos nos polos; (d) arranjo com n pontos no equador

Não foi notada uma diferença sensível nos resultados, a menos de alguns casos especiais, com $n = 47$ onde os pontos distribuidos no equador com uma pequena perturbação levaram a um resultado melhor que os pontos sorteados aleatoriamente sobre a esfera. No entanto, nesse mesmo caso, o tempo de processamento para os pontos sorteados aleatoriamente é quase a metade do tempo de processamento do algoritmo com os pontos inicialmente distribuitos uniformemente no equador e depois perturbados.

Tabela 3 - Escolhendo os pontos iniciais

	RANDOM		EQUATOR + NOISE		POLE + NOISE	
n	f^*	tempo	f^*	tempo	f^*	tempo
3	1.73201	0.01	1.73201	0.01	1.73201	0.01
4	3.67421	0.01	3.8284	0.01	3.8284	0.01
5	6.47457	0.02	6.47479	0.01	6.47457	0.01
6	9.98482	0.01	9.98573	0.01	9.98482	0.01
7	14.4528	0.01	14.4528	0.02	14.4528	0.01
8	19.6748	0.01	19.6748	0.01	19.9488	0.01
9	25.7598	0.03	25.7598	0.02	25.7598	0.01
10	32.7164	0.03	32.7164	0.02	32.7164	0.01

11	40.5953	0.03	40.5953	0.04	40.5953	0.02
12	49.163	0.03	49.1629	0.02	49.163	0.03
13	58.8522	0.04	58.8521	0.04	58.8522	0.05
14	69.3044	0.03	69.3044	0.02	69.3044	0.04
15	80.6668	0.04	80.6668	0.05	80.6668	0.04
16	92.9099	0.05	92.9099	0.06	92.9099	0.04
17	106.047	0.05	106.047	0.04	106.047	0.06
18	120.08	0.04	120.08	0.05	120.08	0.06
19	135.087	0.13	135.087	0.11	135.087	0.13
20	150.878	0.06	150.878	0.1	150.878	0.06
21	167.636	0.1	167.636	0.11	167.636	0.08
22	185.299	0.11	185.279	0.06	185.279	0.07
23	203.925	0.12	203.925	0.07	203.925	0.09
24	223.34	0.08	223.34	0.09	223.34	0.1
25	243.802	0.19	243.802	0.24	243.802	0.21
26	265.127	0.15	265.127	0.15	265.127	0.15
27	287.294	0.11	287.294	0.13	287.294	0.09
28	310.479	0.11	310.479	0.14	310.479	0.11
29	334.627	0.24	334.618	0.31	334.618	0.23
30	359.593	0.13	359.593	0.26	359.593	0.14
31	385.517	0.12	385.517	0.16	385.517	0.16
32	412.243	0.11	412.243	0.1	412.243	0.11
33	440.2	0.4	440.2	0.35	440.2	0.31
34	468.889	0.29	468.889	0.21	468.889	0.23
35	498.57	0.27	498.566	0.46	498.566	0.26
36	529.118	0.57	529.118	0.53	529.118	0.51
37	560.622	0.38	560.622	0.4	560.613	0.24
38	593.026	0.2	593.026	0.17	593.036	0.29
39	626.374	0.19	626.426	0.56	626.374	0.19
40	660.707	0.4	660.707	0.23	660.657	0.2
41	695.895	0.22	695.895	0.23	695.895	0.22
42	732.053	0.27	732.053	0.28	732.053	0.24
43	769.161	0.49	769.161	0.83	769.161	0.39
44	807.139	0.59	807.139	0.41	807.139	0.4
45	846.148	0.77	846.148	0.32	846.148	0.87
46	886.161	0.41	886.161	0.43	886.16	0.42
47	927.129	0.33	927.05	0.63	927.05	0.64
48	968.7	0.41	968.7	0.42	968.7	0.53
49	1011.54	0.32	1011.54	0.51	1011.54	0.39
50	1055.16	0.4	1055.16	0.38	1055.16	0.47

0.3.4 Iterpretando os multiplicadores de Lagrange

A condição de Lagrange $\nabla_x L(x, \lambda) = 0$ tem uma interpretação física particular nesse problema. De fato, pensando que a função objetivo $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ é uma energia potencial $U(x^1, ..., x^n)$, temos que:

$$\nabla_{x^i} L(x, \lambda) = \nabla_{x^i} U(x^1, ..., x^n) + 2\lambda_i x^i = 0$$

Como $-\nabla_{x^i} U(x^1, ..., x^n) = \sum_{j \neq i} F_{ji}$ onde F_{ji} é a força que o elétron j exerce

no elétron i. Temos que a condição Lagrangeana é equivalente à Terceira Lei de Newton. No caso, a componente $-2\lambda_i x^i$ é a força normal que a "superfície esférica"exerce nos elétrons. Sob a ótica da Terceira Lei de Newton, a condição Lagrangeana nos diz que a resultande de forças sobre o elétron (sem contar com a normal da esfera) é perpendicular à superfície, ou seja, não há componente de forças que façam os elétrons deslizarem pela superfície. Pela simetria do problema, imaginamos que os multiplicadores são todos muito parecidos - não parece que uns elétrons façam muito mais força sobre a esfera que outros. E, outra coisa que parece intuitiva é que, nas configurações de equilíbrio, a resultante de forças sobre a esfera seja nula. E, de fato, pelo menos para valores pequenos de n, isso pode ser verificado empiricamente. Para uma prova analítica, basta ver que $F_{ij} = -F_{ji}$ e analisar

$$\sum_i \nabla_{x^i} L(x,\lambda) = \sum_i (-\sum_{j\neq i} F_{ji} + 2\lambda_i x^i) = 2\sum_i \lambda_i x^i = 0$$

Analisemos o vetor de multiplicadores de Lagrange para valores pequenos de n.

$n = 3 \quad \lambda = [0.288666, 0.288653, 0.288677]$
$n = 4 \quad \lambda = [0.45927, 0.459272, 0.459271, 0.459276]$
$n = 5 \quad \lambda = [0.64219, 0.655296, 0.642196, 0.642188, 0.655297]$
$n = 6 \quad \lambda = [0.831986, 0.831986, 0.831997, 0.83199, 0.83199, 0.831997]$

De fato, analisando a média e o desvio padrão do conjunto $\{\lambda_1, ..., \lambda_n\}$ bem como a soma das forças aplicadas sobre a esfera, dada por: $\sum_i \lambda_i x^i$, vemos que a força aplicada sobre a esfera é bem próxima de zero e que os multiplicadores se encontram uns muito próximos aos outros (eles apresentam um desvio padrão baixo), o que mostra que todos os elétrons, no *ground state* exercem uma força semelhante na esfera.

Tabela 4 - Interpretando os multiplicadores

n	media $\{\lambda_i\}$	desvio $\{\lambda_i\}$	$\sum_i \lambda_i x^i$
3	0.288665	9.93233e-06	(1.40564e-05, 9.63922e-06, 3.53419e-06)
4	0.459272	2.27439e-06	(3.29486e-06, 4.76609e-06, -1.6011e-06)
5	0.647433	0.00642029	(4.40766e-06, -3.35451e-05, -3.78374e-06)
6	0.831991	4.79667e-06	(1.94923e-07, 3.06743e-07, -5.17119e-08)
7	1.03231	0.0148424	(-1.3258e-05, 1.6178e-06, 8.94089e-06)
8	1.22961	1.31163e-05	(-5.215e-06, 1.01456e-05, 4.08094e-05)
9	1.43107	0.00603142	(3.91376e-05, -1.6707e-05, -1.53367e-05)
10	1.63577	0.0063065	(-9.55684e-06, -1.49555e-05, 5.08314e-06)
11	1.84513	0.0105327	(-2.36439e-06, 1.21271e-05, 1.91959e-05)
12	2.04827	1.58214e-05	(3.53617e-08, -5.13037e-08, 7.92983e-08)
13	2.26346	0.0129967	(-9.3198e-06, 9.51499e-06, 3.51653e-06)
14	2.47501	0.0137982	(3.6991e-05, 8.15338e-05, 3.54657e-05)
15	2.68866	0.011355	(2.69568e-05, 1.0562e-05, 3.67111e-05)
16	2.9036	0.00686749	(1.10779e-05, -1.12216e-05, -1.72982e-05)
17	3.11887	0.000904371	(-3.17759e-05, -3.01226e-05, 2.65061e-05)

18	3.33529	0.00551578	(-7.16054e-06, 4.22054e-05, 5.04064e-05)
19	3.55478	0.0135263	(4.02236e-05, -3.00393e-05, -9.61125e-06)
20	3.77174	0.00450597	(1.092e-05, 1.16997e-05, 1.47763e-05)
21	3.99104	0.00941977	(-1.2163e-05, -8.08789e-06, -7.03748e-06)
22	4.21047	0.0061627	(-4.26677e-05, 1.85442e-05, -2.05291e-05)
23	4.43293	0.0142952	(-2.41627e-05, 1.35149e-05, -1.44459e-05)
24	4.6526	6.48095e-06	(1.54488e-05, -5.68695e-05, -5.51356e-05)
25	4.87562	0.011919	(1.49523e-05, 4.12644e-05, 2.26505e-05)
26	5.09835	0.012811	(8.18478e-06, -5.78501e-06, 3.39737e-05)
27	5.31992	0.00753546	(-4.16927e-06, 1.52478e-05, 5.95513e-05)
28	5.54384	0.0114888	(6.2717e-06, -1.21002e-05, 4.93664e-05)
29	5.7687	0.0142027	(1.05208e-05, 2.00563e-05, -5.28364e-05)
30	5.99287	0.0090684	(-9.61195e-06, -4.63988e-06, -1.83318e-05)
31	6.21756	0.0126608	(-3.30994e-05, 1.79343e-05, -7.40453e-06)
32	6.44071	0.00696925	(-9.72102e-10, 3.65086e-10, 1.65108e-10)
33	6.66936	0.0144124	(-2.83783e-05, -1.09067e-05, 1.60099e-06)
34	6.89495	0.0141995	(3.57605e-05, 5.1193e-05, 2.88563e-05)
35	7.12227	0.0134099	(4.21759e-05, 8.50675e-05, -3.91532e-05)
36	7.34873	0.0109096	(-1.19141e-06, -1.11604e-06, 4.55496e-05)
37	7.57583	0.0136278	(7.38682e-06, 6.08749e-05, 0.000130507)
38	7.80263	0.00941205	(3.21103e-05, 1.87916e-05, 2.53543e-05)
39	8.03004	0.0158893	(-1.13322e-06, 3.55817e-06, 9.37668e-06)
40	8.25775	0.0154831	(-2.91821e-05, 3.26537e-05, 6.45996e-08)
41	8.486	0.0152257	(2.13519e-05, 1.45644e-05, -2.22571e-05)
42	8.7143	0.0140164	(4.99147e-05, -5.22763e-06, 4.66791e-05)
43	8.94303	0.0138863	(-1.59796e-05, -2.7965e-05, -1.81916e-05)
44	9.17124	0.00959295	(-2.91984e-07, -2.87627e-07, 3.07563e-07)
45	9.40074	0.0107339	(3.17184e-06, 3.79968e-05, 3.94431e-05)
46	9.63202	0.0142658	(-5.15988e-05, 3.08668e-05, 4.61446e-05)
47	9.86198	0.0154084	(-1.62494e-05, -1.03416e-05, -1.25041e-05)
48	10.0903	0.00989432	(-1.22834e-05, -1.0238e-05, 1.84675e-06)
49	10.3215	0.0154804	(-1.10601e-05, 8.44213e-06, 2.81039e-05)
50	10.5513	0.0143822	(2.66926e-05, -2.62792e-05, 1.50821e-05)

0.3.5 Alterando parâmetros

Até esse momento, todos os experimentos foram feitos com RHOMULT = 10 e RHOFRAC = 0.5. Verifiquemos qual o efeito de alterar esses parâmetros. Vamos fazer o teste para $n = 100$ pontos gerados aleatoriamente. Na Tabela 5 encontra-se o resultado de um experimento testando diferentes valores para RHOMULT (que é o γ discutido no início). O valor de RHOFRAC permanece 0.5.

Tabela 5 - Teste com valores diferentes de RHOMULT

RHOMULT	f^*	OI	II	LFE	LGE	tempo
2	4448.45	8	1095	2758	1145	8.12
4	4448.38	6	1241	3356	1292	9.58
8	4448.35	5	886	2582	920	6.71
16	4448.47	5	1835	5483	1926	15.37

32	4448.41	5	1836	6437	1919	12.14
64	4448.33	4	669	1662	698	5.18
128	4448.35	4	278	779	299	2.29
256	4448.35	4	246	681	273	2.04
512	4448.35	4	330	944	352	3.07
1024	4448.35	4	287	783	319	2.56
2048	4448.35	4	480	1185	511	4.08
4096	4448.35	6	301	1175	332	2.47
8192	4448.35	9	300	1432	333	2.88
16384	4448.35	22	695	4194	775	7.12
32768	4448.29	3	308	1183	337	2.52
65536	4448.32	3	309	1171	338	2.55
131072	4448.33	3	317	1291	343	2.47
262144	4448.34	3	366	1574	398	3.03
524288	4448.35	3	395	1938	421	3.53
1048576	4448.35	3	470	2321	506	4.16

Como era esperado, quanto maior o valor de RHOMULT, menor o número de Iterações Exteriores, pois alcançamos a penalização ideal mais rápido. O custo disso, no entanto, é trabalhar com números maiores e portanto mais sujeitos a erros de aproximação e truncamento. O melhor valor para f^* foi conseguido para RHOMULT = 32768. Como em geral ocorre nos problemas práticos, é necessário achar um valor adequado para o parâmetro no tipo específico de problema que se está trabalhando - nem muito grande nem muito pequeno.

No próximo experimento, vamos alterar o valor de RHOFRAC, deixando o valor de RHOMULT constante igual a 10. O resultado desse experimento se encontra documentado na Tabela 6.

Tabela 6 - Teste com valores diferentes de RHOFRAC

RHOFRAC	f^*	OI	II	LFE	LGE	tempo
0.25	4448.35	6	483	1160	517	3.88
0.125	4448.35	6	307	902	336	2.22
0.0625	4448.35	6	284	732	315	2.06
0.03125	4448.35	6	294	862	323	2.23
0.015625	4448.48	5	922	2642	976	6.06
0.0078125	4448.46	5	1301	3736	1359	9.21
0.00390625	4448.4	5	1042	3348	1084	6.31
0.00195312	4448.33	5	1021	2717	1073	8.13
0.000976562	4448.33	5	288	899	308	2.32
0.000488281	4448.33	5	285	914	311	2.14
0.000244141	4448.33	5	286	735	310	2.01
0.00012207	4448.33	5	279	821	304	2.12
6.10352e-05	4448.33	5	264	657	285	2.02
3.05176e-05	4448.33	5	272	722	293	1.99
1.52588e-05	4448.33	5	260	727	283	1.98
7.62939e-06	4448.33	5	279	714	303	1.93
3.8147e-06	4448.39	5	466	1257	494	3.18
1.90735e-06	4448.43	5	1023	2864	1078	7.99
9.53674e-07	4448.33	5	783	2009	823	5.68

Se diminuimos o valor de RHOFRAC, o parametro de penalização ρ é aumentado mais rápidamente, pois se exige que a viabilidae seja conseguida mais rápido. Como podemos ver no experimento da Tabela 5, um valor de RHOFRAC pequeno ajuda o algoritmo a melhorar os resultados e a diminuir o tempo de execução, no entanto, um valor muito pequeno pode ser problemático - forçando o algoritmo a aumentar o parâmetro ρ mesmo quando isso não é necessário. Além do que, trabalhar com valores muito pequenos é sempre bastante sujeito a erros de aproximação.

Outros parâmetros que a implementação do ALGENCAN disponível na TANGO oferece são: GTYPE (que permite que usemos diferenças finitas ao invés de uma função específico para calular o gradiente) e HTYPE (que define o método de cálculo do produto da Hessiana por um vetor). Utilizar diferenças finitas nesse método, não é, de acordo com a documentação da TANGO, barato nem seguro. Portanto não consideraremos opções envolvendo diferenças finitas. Quanto ao valor de HTYPE, comparamos na Tabela 7 os valores obtidos usando as opções 0, 4 e 9 do HTYPE. As demais opções ou não se aplicam ou se reduzem às já mencionadas. De fato, não temos nesse problema restrições de desigualdade nem restrições lineares - que são tratadas em especial em algumas opções.

Tabela 7 - Experimentos com HTYPE

	HTYPE = 0		HTYPE = 4		HTYPE = 9	
n	f^*	tempo	f^*	tempo	f^*	tempo
3	1.73201	0.01	1.73201	0.01	1.73201	0.01
4	3.67421	0.01	3.67421	0.01	3.67421	0.01
5	6.47457	0.01	6.47457	0.01	6.47457	0.01
6	9.98482	0.01	9.98481	0.01	9.98482	0.01
7	14.4528	0.01	14.4528	0.01	14.4528	0.01
8	19.6748	0.01	19.6748	0.02	19.6748	0.01
9	25.7598	0.02	25.7598	0.01	25.7598	0.01
10	32.7164	0.02	32.7164	0.01	32.7164	0.02
11	40.5953	0.01	40.5953	0.02	40.5953	0.02
12	49.163	0.01	49.163	0.02	49.163	0.02
13	58.8522	0.03	58.8522	0.01	58.8522	0.04
14	69.3044	0.03	69.3044	0.02	69.3044	0.03
15	80.6668	0.05	80.6668	0.01	80.6668	0.04
16	92.9099	0.04	92.9099	0.04	92.9099	0.03
17	106.047	0.06	106.047	0.03	106.047	0.06
18	120.08	0.04	120.08	0.04	120.08	0.06
19	135.087	0.06	135.087	0.03	135.087	0.11
20	150.878	0.08	150.878	0.02	150.878	0.08
21	167.636	0.07	167.636	0.05	167.636	0.13
22	185.279	0.08	185.279	0.04	185.279	0.08
23	203.925	0.09	203.925	0.03	203.925	0.1
24	223.34	0.09	223.34	0.06	223.34	0.1
25	243.802	0.14	243.802	0.05	243.802	0.17
26	265.127	0.12	265.127	0.06	265.127	0.18
27	287.294	0.15	287.294	0.04	287.294	0.24

28	310.479	0.12	310.479	0.04	310.479	0.12
29	334.627	0.18	334.618	0.07	334.627	0.15
30	359.593	0.19	359.593	0.08	359.593	0.26
31	385.517	0.15	385.517	0.05	385.517	0.16
32	412.243	0.15	412.243	0.05	412.243	0.32
33	440.195	0.2	440.2	0.06	440.195	0.27
34	468.889	0.2	468.889	0.08	468.889	0.24
35	498.566	0.26	498.566	0.08	498.566	0.34
36	529.118	0.26	529.118	0.07	529.118	0.5
37	560.622	0.26	560.622	0.09	560.622	0.39
38	593.036	0.32	593.036	0.1	593.026	0.81
39	626.426	0.3	626.426	0.09	626.426	0.48
40	660.657	0.39	660.657	0.09	660.657	0.37
41	695.895	0.22	695.895	0.09	695.895	0.31
42	732.053	0.3	732.053	0.11	732.053	0.32
43	769.161	0.47	769.161	0.09	769.161	0.66
44	807.139	0.39	807.139	0.09	807.139	0.5
45	846.148	0.48	846.148	0.1	846.148	0.55
46	886.16	0.46	886.16	0.15	886.16	0.76
47	927.06	0.59	927.06	0.15	927.06	1.15
48	968.7	0.46	968.7	0.12	968.7	0.7
49	1011.54	0.38	1011.54	0.14	1011.54	0.7
50	1055.16	0.5	1055.16	0.15	1055.16	0.46
51	1099.8	0.54	1099.8	0.18	1099.8	1.22
52	1145.39	0.53	1145.39	0.13	1145.39	2.04
53	1191.9	0.64	1191.9	0.19	1191.89	2.82
54	1239.33	0.72	1239.33	0.18	1239.33	0.68
55	1287.74	0.76	1287.74	0.19	1287.74	0.51
56	1337.05	0.72	1337.05	0.17	1337.05	0.62
57	1387.34	0.57	1387.34	0.22	1387.34	0.71
58	1438.57	0.7	1438.57	0.19	1438.57	0.5
59	1490.73	0.77	1490.73	0.27	1490.73	0.7
60	1543.77	0.82	1543.77	0.23	1543.77	1.08
61	1597.88	0.7	1597.88	0.24	1597.88	0.97
62	1652.85	0.98	1652.85	0.2	1652.85	0.88
63	1708.86	1.54	1708.86	0.28	1708.86	1.74
64	1765.79	1.22	1765.79	0.33	1765.79	1.3
65	1823.65	1.01	1823.65	0.34	1823.65	0.94
66	1882.42	1.6	1882.42	0.27	1882.42	2.39
67	1942.1	1.01	1942.1	0.25	1942.1	0.7
68	2002.86	1.01	2002.86	0.34	2002.86	1.08
69	2064.51	1.23	2064.51	0.36	2064.51	1.5
70	2127.08	1.33	2127.08	0.41	2127.08	1.17
71	2190.61	2.17	2190.61	0.41	2190.61	2.41
72	2254.96	1.03	2254.96	0.36	2254.96	1.59
73	2320.59	4.18	2320.59	4.93	2320.59	13.98
74	2387.04	1.95	2387.04	1.3	2387.04	7.47
75	2454.31	2.45	2454.31	1.62	2454.31	7.59
76	2522.61	4.67	2522.61	8.21	2522.61	19.91
77	2591.84	4.49	2591.84	5.64	2591.84	22.01

78	2662.03	5.8	2661.97	2.98	2662.03	21.46
79	2733.23	5.41	2733.23	3.79	2733.23	33.02
80	2805.34	4.6	2805.42	6.9	2805.34	32.79

A opção HTYPE = 4 é, de acordo com a Tabela 6, a menos custosa computacionalmente. Usando essa opção, o tempo de processamento é, em geral, menor. No entanto, em alguns casos, os resultados não são tão bons quanto àqueles produzidos pelas outras duas opções (0 e 9). Isso é bem ilustado pelo caso $n = 78$. Isso ocorre pois com essa opção (HTYPE = 4) evitamos recalcular a Hessiana a cada iteração. A opção (HTYPE = 0) é mais custosa que a opção (HTYPE = 4) pois temos que recalcular a hessiana em cada ponto. A opção (HTYPE = 9) se mostrou mais ineficiente que a (HTYPE = 0) na maioria dos caso, mas produzindo resultados tão bons quanto esta.

0.4 Variações do problema

0.4.1 Fixando um ponto

Se $\Gamma \in O_3(\mathbb{R})$, ou seja, se é uma transformação ortogonal ($\Gamma\Gamma^\top = \Gamma^\top\Gamma = I$) então se $f(\Gamma x^1, ..., \Gamma x^n) = f(x^1, ..., x^n)$, de fato, f só depende de $\|x^i - x^j\|$ e, temos:

$$\|\Gamma x^i - \Gamma x^j\|^2 = (x^i - x^j)^\top \Gamma^\top \Gamma (x^i - x^j) = (x^i - x^j)^\top (x^i - x^j) = \|x^i - x^j\|^2$$

Assim, dada uma solução (global ou local), podemos achar uma infinidade de outras soluções soluções pela ação do grupo $O_3(\mathbb{R})$. Assim, faz sentido pensar no espaço de possíveis soluções para esse problema módulo a relação de equivalência induzida no $\mathbb{R}^3$ pela ação desse grupo.

Outro grupo a considerar é o próprio S_n - o grupo das permutações de n elementos. De fato, dada uma solução $x = (x^1, ..., x^n)$ e $\sigma \in S_n(\{1, ..., n\})$ temos que $x^\sigma := (x^{\sigma(1)}, ..., x^{\sigma(n)})$ também é uma solução do problema.

Resolver o problema de considerar as soluções módulo S_n é mais complicado, pois ele tem um caráter mais combinatório. No entanto, diminuir o espaço de soluções (no que diz respeito à ação de $O_3(\mathbb{R})$ é mais simples. Uma forma de fazer isso, é fixar um ponto (isso ainda não toma o espaço módulo $O_3(\mathbb{R})$ - isso só diminui o espaço de possíveis soluções). Se, dado um p fixo sobre a esfera, obrigamos que: $x^0 = p$, já diminuimos o espaço de possíveis soluções em 2 dimensões - e note que não descartamos nenhuma solução, pois dada qualquer solução, x, sempre existe um $\Gamma \in O_3(\mathbb{R})$ que leva x^0 em p. Com isso temos 3 dimensões e 1 restrição a menos, ou seja, 2 dimensões (no sentido topológico).

Fixando um segundo ponto não temos mais um espaço de soluções equivalente ao original, mas o que podemos fazer é fixar uma geodésica passando pelo primeiro ponto fixado onde o segundo ponto deve estar. Vejamos como isso afeta o algoritmo. Vamos fixar um ponto, e considerar o problema agora em $\mathbb{R}^{3(n-1)}$ com:

$$f_1(x) = \sum_{i=1}^{n-1} \sum_{j=1}^{i-1} \frac{1}{\|x^i - x^j\|} + \sum_{i=1}^{n-1} \frac{1}{\|x^i - p\|}$$

Tabela 8 - Um ponto fixado

n	f^*	OI	II	LFE	LGE	tempo
3	1.73203	3	14	42	22	0.01
4	3.67422	4	24	58	36	0
5	6.47462	4	42	81	57	0.01
6	9.98499	4	26	45	36	0.01
7	14.4529	5	99	267	116	0.03
8	19.6749	5	69	124	87	0.02
9	25.7591	5	86	174	107	0.04
10	32.7166	6	74	127	92	0.01
11	40.5956	6	107	201	127	0.04
12	49.1635	6	68	116	84	0.03
13	58.8524	7	173	350	199	0.06
14	69.3048	7	84	140	102	0.04
15	80.6676	7	160	307	183	0.08
16	92.9103	8	135	226	157	0.08
17	106.048	8	229	490	259	0.15
18	120.081	8	163	312	188	0.09
19	135.084	8	357	791	389	0.25
20	150.878	9	119	209	146	0.12
21	167.637	9	314	679	346	0.25
22	185.28	9	177	322	203	0.13
23	203.926	10	177	325	205	0.14
24	223.341	10	206	384	233	0.16
25	243.804	10	1245	3032	1323	1.11
26	265.128	11	302	615	334	0.35
27	287.295	11	210	407	241	0.19
28	310.481	11	265	524	297	0.24
29	334.621	11	522	1215	566	0.53
30	359.595	12	466	1015	514	0.47
31	385.519	12	211	385	246	0.21
32	412.246	12	154	283	184	0.19
33	440.184	12	1177	2808	1250	1.57
34	468.898	9	471	961	501	0.56
35	498.553	8	369	772	395	0.58
36	529.118	8	1195	2752	1256	1.66
37	560.623	8	407	821	443	0.58
38	593.027	7	358	694	389	0.54
39	626.375	7	274	536	298	0.45
40	660.659	7	320	623	345	0.61
41	695.897	7	270	500	294	0.49
42	732.055	7	214	378	235	0.41
43	769.164	7	526	1176	562	1.21
44	807.143	7	877	1988	922	2.08

45	846.181	7	880	2117	922	1.92
46	886.166	8	615	1396	653	1.25
47	927.056	8	844	1938	881	1.85
48	968.701	7	515	1117	545	1.08

De fato, o resultado dos valores ótimos foi muito semelhante àquele obtido na Tabela 1, no entanto o esforço computacional aumentou bastante. O tempo de processamento nesse experimento foi, em média, o dobro do tempo de processamento no experimento anterior. Na Figura 7, comparamos o número de vezes que o Lagrangeano foi avaliado em cada caso.

A restrição imposta, mesmo diminuindo a dimensão do espaço de soluções do problema de $2n$ para $2(n-1)$, não apresentou ganho de eficiência. De certa forma, essa restrição extra deu menos liberdade ao algoritmo para explorar o espaço de possíveis soluções e isso foi responsável pela diminuição de eficiência.

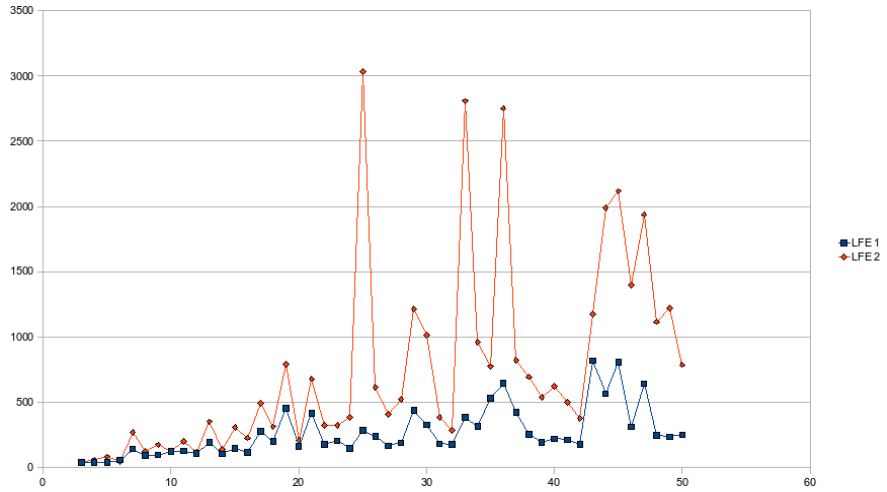


Figura 6: Comparação do numero de avaliações do Lagrangeano Aumentado no algoritmo com um ponto fixado (LFE 2) com o algoritmo sem nenhum ponto fixado (LFE 1)

0.4.2 Tornando o problema irrestrito

Considerando uma parametrização da esfera $g : \mathbb{R}^2 \longrightarrow \mathbb{S}^2 \subseteq \mathbb{R}^3$, podemos levar o problema (1) para um problema irrestrito em $\mathbb{R}^{2n}$. Consideramos então o problema de minimizar a função $f_2(y^1, ..., y^n) = f(g(y^1), ..., g(y^n))$ onde $y^i \in \mathbb{R}^2$. Assim, o gradiente é então dado por:

$$\nabla_{y^i} f_2(y) = \nabla g(y^i) \cdot \nabla_{x^i} f(x)$$

onde $x^i = g(y^i)$ e $\nabla g(y^i) = Jg(y^i)^\top$. Calculando as soluções com a opção HTYPE = 4, i.e, usando aproximações do tipo BFGS para a Hessiana, usando a parametrização descrita em (6), com $y = (\theta, \phi)$. Os resultados se encontram na Tabela 9. Comparando esses resultados com a Tabela 1 em que foi usado

o Método do Lagrangeano Aumentado para o problema tal como descrito em (1), vemos que o tempo de execução diminuiu consideravelmente na maioria dos casos, mas, em compensação os valores encontrados não são tão bons quanto aqueles na Tabela 1 em alguns casos. Voltamos a mesma questão discutida na seção anterior: restringimos o espaço de busca do Algoritmo, tornando-o mais propenso a cair em mínimos locais. Agora, o algoritmo percorre os valores na esfera. Anteriormente, ele podia percorrer o espaço de soluções passando por pontos que não eram viáveis, tentando convergir a pontos viáveis que fossem estacionários.

Para resolver o problema irrestrito usando o GENCAN, temos que especificar os limites das variáveis - definimos a região viável tipicamente como uma caixa. No experimento da Tabela 9, fizemos um experimento sem caixa (ou seja, com o algoritmo tendo liberade para explorar todo o $\mathbb{R}^2$ e com as caixas $[0, 2\pi] \times [0, \pi]$ e $[-2\pi, 4\pi] \times [-\pi, 2\pi]$. O problema da caixa $[0, 2\pi] \times [0, \pi]$ é que ela não captura adequadamente a topologia da esfera (uma devantagem clara desses métodos) - por exemplo, não fica claro que pontos com θ próximo de zero e pontos com θ próximos a 2π são próximos. Aumentando o tamanho da caixa, capturamos de alguma forma essas características topológicas - e isso, de fato, leva a resultados melhores em alguns casos, por exemplo, o caso $n = 21$, mesmo sob pena de um maior esforço computacional.

Para outros casos (como $n = 38$) o melhor resultado ocorreu para o problema sem caixa, i.e., $y \in \mathbb{R}^2$.

Tabela 9 - Resolvendo o problema irrestrito

	$y \in \mathbb{R}^2$		$y \in [0, 2\pi] \times [0, \pi]$		$y \in [-2\pi, 4\pi] \times [-\pi, 2\pi]$	
n	f^*	tempo	f^*	tempo	f^*	tempo
3	1.73205	0.01	1.73205	0.01	1.73205	0.01
4	3.67423	0.01	3.67423	0.01	3.82843	0.01
5	6.47469	0.01	6.47469	0.01	6.47469	0.01
6	9.98528	0.01	9.98528	0.01	9.98528	0.01
7	14.453	0.01	14.453	0.01	14.453	0.01
8	19.6753	0.01	19.6753	0.01	19.6753	0.01
9	25.76	0.01	25.76	0.01	35.9154	0.01
10	32.7169	0.01	32.7169	0.01	32.7169	0.01
11	40.5965	0.01	40.5966	0.03	40.5965	0.01
12	49.1653	0.02	49.1653	0.02	49.1653	0.01
13	58.8532	0.03	58.8532	0.05	58.8532	0.02
14	69.3064	0.02	69.3064	0.01	69.3064	0.04
15	80.6702	0.01	80.6702	0.05	80.6702	0.03
16	92.9204	0.01	92.9204	0.05	92.9117	0.03
17	106.05	0.02	106.05	0.06	106.05	0.04
18	120.084	0.04	120.084	0.04	120.084	0.16
19	135.089	0.05	135.091	0.06	135.089	0.09
20	150.882	0.04	150.882	0.05	150.882	0.08
21	167.642	0.07	169.66	0.01	167.642	0.24
22	185.288	0.04	185.288	0.05	185.288	0.03
23	203.93	0.04	203.93	0.07	203.93	0.05
24	223.347	0.03	223.395	0.13	223.348	0.07
25	243.813	0.08	243.813	0.15	243.813	0.08

26	265.133	0.08	265.136	0.14	265.133	0.08
27	287.303	0.05	287.303	0.04	287.303	0.07
28	310.492	0.06	310.492	0.18	310.492	0.1
29	334.634	0.14	334.636	0.12	334.634	0.19
30	359.604	0.18	359.604	0.11	359.606	0.09
31	385.531	0.1	385.531	0.41	385.531	0.73
32	412.261	0.05	412.261	0.07	412.261	0.05
33	440.204	0.15	440.204	0.4	440.204	0.17
34	468.905	0.11	468.905	0.33	468.905	0.18
35	498.57	0.12	498.573	0.29	498.57	0.42
36	529.122	0.22	529.122	0.37	529.122	0.38
37	560.628	0.16	560.619	0.1	560.619	0.1
38	593.039	0.08	593.049	0.23	593.049	0.49
39	626.389	0.08	626.389	0.22	626.396	0.12
40	660.675	0.1	660.675	0.38	660.675	0.28
41	695.917	0.1	732.286	0.01	695.917	0.45
42	732.078	0.12	732.078	0.52	732.078	0.17
43	769.191	0.8	769.191	1.66	769.194	0.16
44	807.174	0.29	807.176	0.22	807.174	0.35
45	846.188	0.72	846.188	1.04	15.7396	0.06
46	886.17	0.23	886.167	0.38	886.17	0.41
47	927.062	0.3	927.059	0.28	742.911	0.01
48	968.713	0.19	968.713	0.48	968.713	0.31
49	1011.56	0.12	1011.56	0.19	1011.56	0.69
50	1055.18	0.17	1055.18	0.15	1055.18	0.14

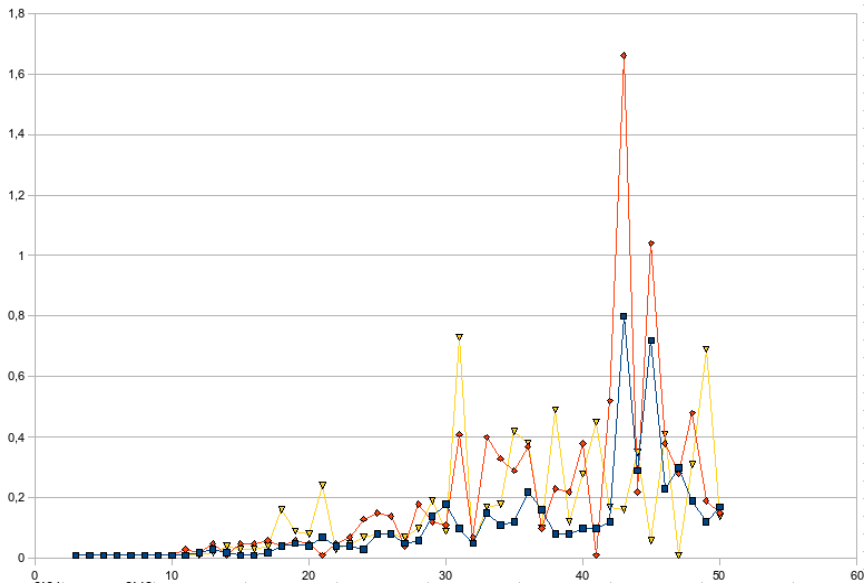


Figura 7: Comparação do tempo de execução para o problema irrestrito com $y \in \mathbb{R}^2$ (azul), $y \in [0, 2\pi] \times [0, \pi]$ (vermelho) e $y \in [-2\pi, 4\pi] \times [-\pi, 2\pi]$ (amarelo)

0.5 Conclusão

O problema (1) é um problema com várias dimensões e com vários mínimos locais - logo constitui um bom campo de testes para algoritmos de otimização - ele mede bem a capacidade dos métodos de evitar mínimos locais e buscar mínimos globais. Ele admite função objetivo e restrições suficientemente simples para que possam ser escritas explicitamente com seus gradientes e hessianas mas ao mesmo tempo suficientemente complicadas para que não seja possível calcular, de forma fácil, o mínimo global analiticamente.

Testamos o método ALGENCAN (particulamente a implementação da TANGO) nesse problema. Conseguimos bons resultados - alguns melhores do que os conseguidos em [HSS97]. Foram testadas várias configurações de parâmetros, variando-se RHOMULT, RHOFRAC, HTYPE, GTYPE, dentre outros. Além disso, foram testadas outras formulações do problema, a saber, o problema com um dos pontos fixados e o problema no espaço de uma parametrização da esfera.

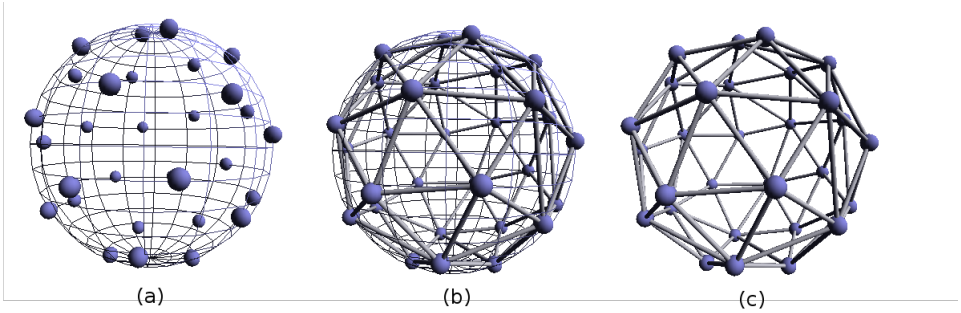


Figura 8: (a) Solução do Problema para $n = 30$; (b) Arestas do fecho convexo dos pontos $\{x^1, ..., x^n\}$ da solução; (c) Mesmo de (b) só que sem a esfera desenhada

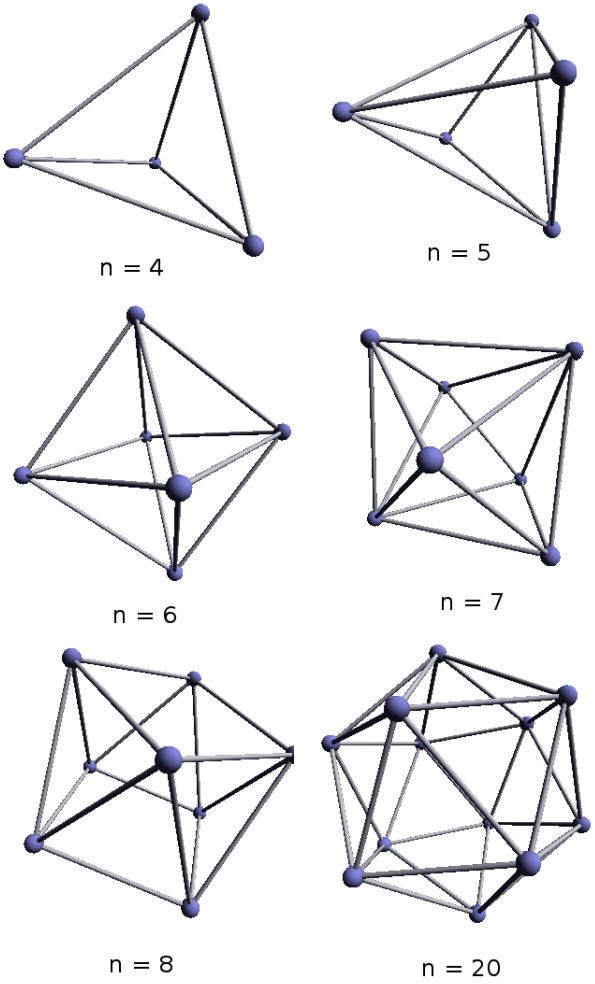


Figura 9: Alguma soluções do problema de Thomson geradas usando OpenGL: ($n = 4$) Tetraedro, ($n = 5$) Di-piramide triangular, ($n = 6$) Octaedro, ($n = 7$) Di-piramide pentagonal, ($n = 8$) Anti-prisma quadrado, ($n = 20$) Icosaedro

Referências Bibliográficas

[ABMS05] Andreani, Birgin, Martinez, and Schuverdt. Augmented lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111:5–32, 2005.

[ABMS07] Andreani, Birgin, Martinez, and Schuverdt. On augmented lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18:1286–1309, 2007.

[Ber99] Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[BM02] Birgin and Martinez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23:101–125, 2002.

[HSS97] Hardin, Sloane, and Smith. Minimal energy arrangements of points on a sphere. *www.research.att.com/∼njas/electrons/*, 1997.