

Análise das Rotinas de Intensificação e Diversificação no Algoritmo Inspirado em Organismos Simbióticos

Wesklei Migliorini*, Leanderson André†, Rafael Stubs Parpinelli†

Universidade do Estado de Santa Catarina

Joinville - SC - Brasil

* Departamento de Ciência da Computação

Email: wesklei.m@gmail

† Programa de Pós-graduação em Computação Aplicada

Email: leanderson@gmail.com, rafael.parpinelli@udesc.br

Resumo—A Natureza é uma fonte inesgotável de inspiração para o desenvolvimento de novas tecnologias. Entre os diferentes fenômenos naturais, as relações simbióticas inspirou recentemente o desenvolvimento de uma abordagem para otimização de problemas complexos. Chamada de Algoritmo de Busca por Organismos Simbióticos (*Symbiotic Organisms Search*, SOS), a abordagem se inspira nas relações de mutualismo, comensalismo e parasitismo. Sabe-se que, para um algoritmo ser bem sucedido no processo de otimização, é necessário que este seja capaz de balancear de maneira eficiente o uso das rotinas de intensificação e diversificação durante o processo de busca. Este trabalho tem como objetivo analisar o impacto dos componentes simbióticos do SOS no seu desempenho e, conseqüentemente, verificar quais destes contribuem para a intensificação e quais contribuem para a diversificação. Para isto, o desempenho do algoritmo SOS foi comparado com o desempenho do seu uso com as combinações possíveis dos componentes simbióticos. Experimentos foram realizados com 17 funções matemáticas contínuas com alta dimensionalidade (30 e 200). Os resultados obtidos indicam que, ao invés de utilizar todas as relações propostas pelo SOS canônico, a utilização das relações de comensalismo e parasitismo apresentam um maior equilíbrio entre intensificação e diversificação, refletindo em melhores resultados.

Keywords—Algoritmo de busca, Intensificação e diversificação, Relações simbióticas, Otimização contínua, Algoritmo bio-inspirado

I. INTRODUÇÃO

A Computação Natural é uma área da Ciência da Computação que utiliza a Natureza como fonte de inspiração e tem como objetivo desenvolver algoritmos baseados em comportamentos naturais [1]. Como exemplo, alguns algoritmos amplamente utilizados na literatura são a Evolução Diferencial [2], Otimização por Colônia de Formigas [3] e Algoritmo de Busca por Organismos Simbióticos (SOS) [4] e a Otimização por Enxame de Partículas [5]. Os algoritmos da Computação Natural são utilizados com sucesso para resolver diversos problemas complexos de otimização do mundo real.

Um algoritmo pode ser bem sucedido na otimização de um determinado problema se ele for capaz de prover um bom equilíbrio no uso dos seus componentes de diversificação e intensificação [6]. Este equilíbrio depende de vários fatores como: a superfície de busca da função objetivo, o

número máximo de avaliações e o tamanho da população. A diversificação refere-se a exploração global, ou seja, explorar o máximo possível do espaço de busca e identificar áreas promissoras com soluções de alta qualidade. A intensificação refere-se a exploração local, mais intensiva em uma área específica do espaço de busca. Um algoritmo com rotinas de diversificação em excesso se aproxima de um processo aleatório. Já o favorecimento em excesso de rotinas de intensificação leva a uma maior chance de estagnar a busca em um ponto sub-ótimo, perdendo a diversidade das soluções. O equilíbrio adequado entre as duas etapas permite explorar com mais eficiência o espaço de busca, e conseqüentemente, obter resultados melhores [7].

Uma metaheurística recente na literatura, o Algoritmo de Busca por Organismos Simbióticos (*Symbiotic Organisms Search*, SOS)[4], tem apresentado um desempenho muito satisfatório em problemas de domínio contínuo. O SOS é inspirado nas relações simbióticas realizadas pelos organismos na Natureza. A simbiose corresponde às interações que os organismos realizam uns com os outros e que afetam direta ou indiretamente os envolvidos. O algoritmo utiliza três relações simbióticas: o mutualismo, o comensalismo e o parasitismo. Estas relações são aplicadas sequencialmente para gerar novas soluções candidatas. A estrutura do algoritmo é simples, sendo livre de parâmetros. Apesar da simplicidade, o SOS obteve resultados promissores e melhores do que outros algoritmos estado da arte em otimização de funções matemáticas contínuas e em problemas de engenharia [4].

Na literatura, outros trabalhos aplicam o algoritmo SOS a problemas do mundo real. Em [8], o SOS é aplicado a um problema contínuo através de uma proposta chamada SOS-Rede Neural Quântica (SOS-QNN) para predição da resistência à compressão de concreto. O SOS-QNN é comparado com um algoritmo de Redes Neurais Artificiais e Redes Neurais Quânticas. Resultados mostraram que a proposta SOS-QNN obteve melhor desempenho indicando ser um algoritmo eficiente para esta classe de problemas. Em [9], o SOS é adaptado e aplicado em um problema discreto para realizar alocação de múltiplos recursos entre vários projetos afim de diminuir o desperdício destes. Os experimentos realizados comparam o SOS com a Evolução Diferencial, Otimização por

Partículas e Algoritmos Genéticos. Resultados experimentais e testes estatísticos mostraram que o SOS obteve melhores resultados. Outro problema discreto é analisado em [10] e o SOS é aplicado para otimizar a alocação de *buffer* em um sistema de manufatura especificado como uma rede de filas fechadas. O algoritmo realiza a maximização da taxa de vazão do sistema. Experimentos mostraram-se promissores, com resultados bastante competitivos.

Dos bons resultados relatados na literatura, percebe-se um forte indício de que o algoritmo SOS possui um bom equilíbrio entre os rotinas de intensificação e diversificação. Porém, verifica-se também que uma análise mais detalhada se faz necessária para compreender qual o impacto dos componentes simbióticos no processo de otimização. Portanto, este artigo tem como objetivo verificar como as relações simbióticas influenciam no resultado final da otimização e como elas contribuem no balanceamento entre a intensificação e a diversificação do processo de busca.

A verificação da influência das relações simbióticas foi realizada através de experimentos fatoriais, utilizando todas as combinações possíveis das relações simbióticas. O experimento realizado utiliza 17 funções *benchmark* contínuas com alta dimensionalidade (30 e 200 dimensões) [11]. Optou-se por uma alta dimensionalidade nas funções para poder avaliar o desempenho dos algoritmos em problemas com alto nível de complexidade. Dois testes estatísticos não-paramétricos são utilizados na análise dos resultados: *Friedman Rank Sum Test* para comparar o desempenho entre todos os algoritmos, e *Wilcoxon Signed Rank Test* para o teste pareado entre os algoritmos [12].

O trabalho está dividido da seguinte forma: a Seção II apresenta uma revisão bibliográfica sobre os conceitos que envolvem o algoritmo SOS; a Seção III discute a contribuição das relações simbióticas presente no SOS para a diversificação e intensificação; a Seção IV apresenta os experimentos realizados e a análise dos resultados; por fim, as conclusões do trabalho são apresentadas na Seção V, juntamente com os trabalhos futuros.

II. ALGORITMO DE BUSCA POR ORGANISMOS SIMBIÓTICOS

O algoritmo de Busca por Organismos Simbióticos (*Symbiotic Organisms Search*) [4] é um algoritmo populacional que se baseia em três relações simbióticas existentes na Natureza: mutualismo, comensalismo e parasitismo. A relação de mutualismo ocorre quando os organismos envolvidos se beneficiam. Esta relação ocorre, por exemplo, entre os gados e os pássaros. Os pássaros se alimentam dos carrapatos que estão sugando o sangue do gado, e consequentemente, o gado se livra dos parasitas. A relação de comensalismo ocorre quando um organismo é beneficiado e outro organismo envolvido não é beneficiado ou não é prejudicado. Este comportamento pode ser observado na relação entre os tubarões e as rêmoras. As rêmoras se alimentam dos restos de alimentos deixados pelos tubarões que recebem nada em troca. A relação de parasitismo ocorre quando o organismo tira proveito de um organismo e o prejudica. Esta relação ocorre entre os gados e carrapatos, orquídeas e árvores, entre outros. Os carrapatos se instalam nos gados e sugam seu sangue, da mesma maneira ocorre com

as orquídeas, que sugam a seiva das árvores em que estão instaladas. Esta relação pode até levar o organismo a morte.

O SOS utiliza uma população de organismos que representa um ecossistema e cada organismo representa uma solução candidata. Cada organismo é avaliado e recebe um valor associado ao problema sendo otimizado que representa seu *fitness*. Neste ecossistema, cada um dos organismos é capaz de realizar as relações de mutualismo, comensalismo e parasitismo. Na relação de mutualismo, um organismo se associa com um outro escolhido aleatoriamente a fim de cooperar e melhorar os seus respectivos *fitness*. O *fitness* representa o nível de adaptação do organismo, ou seja, quanto mais bem adaptado, melhor é a solução para o problema sendo otimizado. Na relação de comensalismo, o organismo se associa com um outro, porém apenas um deles é beneficiado com a relação. Já na relação de parasitismo, o organismo se duplica e sua cópia se torna um parasita que procura obter o lugar de outro organismo dentro da população. Nesta relação, o organismo de maior *fitness* sobrevive. Os detalhes de implementação do SOS são apresentados no Algoritmo 1 considerando problemas de minimização.

O Algoritmo 1 define o tamanho da população com o parâmetro ECO_{size} , a condição de parada pelo número de avaliações da função objetivo $EVAL_{max}$, o limite inferior e superior da função por LB e UB , respectivamente, e a dimensão do problema por DIM . O algoritmo se inicia gerando indivíduos aleatórios que farão parte da população inicial do ecossistema (entre linhas 2 e 4). O laço principal do algoritmo (linha 5) define a condição de parada para o número de avaliações da função de *fitness*. O laço populacional (entre linhas 6 e 32) irá aplicar as relações simbióticas para cada indivíduo da população. O laço populacional inicia-se com a seleção do melhor indivíduo \vec{M} (linha 7) que fará parte das etapas simbióticas de mutualismo e comensalismo. O mutualismo inicia selecionando um indivíduo \vec{X}_j aleatório (linha 8) que irá realizar o mutualismo junto a \vec{X}_i . O vetor *Mutual* é gerado e representa as características de relacionamento entre os vetores \vec{X}_i e \vec{X}_j (linha 9). Os valores de BF_1 e BF_2 correspondem aos valores 1 ou 2, atribuídos de forma aleatória e representam o valor de benefício mutualístico entre os dois indivíduos. Quanto maior o valor, mais benéfica é a relação. Utilizando as variáveis anteriores, os novos indivíduos \vec{X}'_i e \vec{X}'_j são gerados (linhas 11 e 12). É importante observar que este processo consome duas avaliações da função objetivo (linhas 13 e 14). As novas soluções são avaliadas e caso sejam melhores que \vec{X}_i e \vec{X}_j atuais, são aceitas como novas soluções (linhas 15 até 18). A etapa de comensalismo inicia-se selecionando um indivíduo \vec{X}_j (linha 19) que irá realizar o comensalismo mas, diferentemente do mutualismo, apenas \vec{X}_i recebe influência do processo e \vec{X}_j não sofre qualquer alteração. A nova solução \vec{X}'_i é gerada podendo aproximar ou distanciar \vec{X}_i da solução \vec{M} e \vec{X}_j (linha 20). A relação de comensalismo é avaliada e caso \vec{X}'_i seja melhor, o valor de \vec{X}_i é substituído por ela (entre linhas 21 e 24). A etapa de parasitismo inicia (linha 25) selecionando um indivíduo \vec{X}_j que serve como um hospedeiro para o vetor parasita e poderá ser substituído da população. A partir do organismo \vec{X}_i é criado um vetor parasita \vec{P} (linha 26), que é uma cópia de \vec{X}_i . Uma das dimensões de \vec{P} é modificada por um número aleatório k dentro do intervalo LB e UB da função

objetivo (linha 28). O parasita \vec{P} é avaliado (linhas 29 até 32) em relação a \vec{X}_j , substituindo \vec{X}_j caso tenha melhor valor de solução. Ao final do algoritmo a melhor solução encontrada é retornada (linha 34).

Algoritmo 1: Esquema geral do algoritmo SOS

```

1
  input :  $ECO_{size}, EVAL_{max}, LB, UB, DIM$ 
  output: Melhor solução encontrada  $\vec{s}$ 

  // Inicialização do Ecosistema
2 for ( $i=0; i < ECO_{size}; i++$ ) do
3   Inicializar  $\vec{X}_i$  aleatoriamente
4   Avaliar função objetivo  $f(\vec{X}_i)$ 

5 while ( $eval < EVAL_{max}$ ) do
  // Etapa populacional
6 for ( $i=0; i < ECO_{size}; i++$ ) do
7    $\vec{M} \leftarrow$  Melhor indivíduo no ecossistema

  // Etapa de Mutualismo
8    $\vec{X}_j \leftarrow$  indivíduo aleatório,  $i \neq j$ 
9    $Mutual \leftarrow \frac{\vec{X}_i + \vec{X}_j}{2}$ 
10   $BF_1, BF_2 \leftarrow$  1 ou 2 aleatório
11   $\vec{X}'_i = \vec{X}_i + rand(0, 1) * (\vec{M} - Mutual * BF_1)$ 
12   $\vec{X}'_j = \vec{X}_j + rand(0, 1) * (\vec{M} - Mutual * BF_2)$ 
13  Avaliar  $f(\vec{X}'_i)$  e  $f(\vec{X}'_j)$ 
14   $eval += 2$ 
15  if  $f(\vec{X}'_i) < f(\vec{X}_i)$  then
16     $\vec{X}_i \leftarrow \vec{X}'_i$ 
17  if  $f(\vec{X}'_j) < f(\vec{X}_j)$  then
18     $\vec{X}_j \leftarrow \vec{X}'_j$ 

  // Etapa de Comensalismo
19   $\vec{X}_j \leftarrow$  indivíduo aleatório,  $i \neq j$ 
20   $\vec{X}'_i = \vec{X}_i + rand(-1, 1) * (\vec{M} - \vec{X}_j)$ 
21  Avaliar  $f(\vec{X}'_i)$ 
22   $eval ++$ 
23  if  $f(\vec{X}'_i) < f(\vec{X}_i)$  then
24     $\vec{X}_i \leftarrow \vec{X}'_i$ 

  // Etapa de Parasitismo
25   $\vec{X}_j \leftarrow$  indivíduo aleatório,  $i \neq j$ 
26   $\vec{P} = \vec{X}_i$ 
27   $k \leftarrow random(0, DIM)$ 
28   $P_k \leftarrow random(LB, UB)$ 
29  Avaliar  $f(\vec{P})$ 
30   $eval ++$ 
31  if  $f(\vec{P}) < f(\vec{X}_j)$  then
32     $\vec{X}_j \leftarrow \vec{P}$ 

33  $\vec{s} \leftarrow$  Encontrar melhor solução do ecossistema
34 return  $\vec{s}$ 

```

III. PAPEL DAS RELAÇÕES SIMBIÓTICAS COMO ROTINAS DE INTENSIFICAÇÃO E DIVERSIFICAÇÃO

A intensificação e a diversificação são duas etapas fundamentais no processo de otimização [7]. O equilíbrio adequado entre as duas etapas permite explorar com mais eficiência o espaço de busca, e conseqüentemente, obter resultados melhores. Portanto, para equilibrar a intensificação e a diversificação é necessário identificar quais componentes contribuem para cada uma das etapas. O SOS é composto por três componentes simbióticos: mutualismo, comensalismo e o parasitismo. Analisando o algoritmo SOS é possível verificar que as relações de mutualismo e comensalismo são responsáveis pela intensificação da busca enquanto a relação de parasitismo é responsável pela diversificação.

O mutualismo utiliza a diferença da melhor solução (\vec{M}) com a combinação de duas soluções candidatas (\vec{X}_i e \vec{X}_j) para gerar uma nova solução. A influência da relação de mutualismo na nova solução é determinada por um valor aleatório com distribuição uniforme no intervalo de $[0, 1]$. O objetivo dessa relação é deslocar as duas soluções \vec{X}_i e \vec{X}_j para a mesma região do espaço de busca onde \vec{M} está localizado. O tamanho do deslocamento no espaço de busca depende da cooperação entre as duas soluções e é influenciado diretamente pela diversidade. Se a diversidade é alta, o deslocamento no espaço de busca é grande. Se a diversidade é pequena, o deslocamento é pequeno e quando não há diversidade, conseqüentemente, não há deslocamento. Uma vez que a diversidade da população é perdida durante o processo de otimização, a relação de mutualismo não é capaz de repará-la, pois depende da diversidade da população para gerar novas soluções. Portanto, com uma população sem diversidade, a relação de mutualismo não realiza perturbações nas soluções \vec{X}_i e \vec{X}_j .

A relação de comensalismo utiliza a diferença da melhor solução (\vec{M}) com uma solução candidata (\vec{X}_j) para gerar uma nova solução. A principal diferença entre a relação de comensalismo e mutualismo está na influência da relação na nova solução gerada. A influência do comensalismo é determinada por um valor aleatório entre $[-1, 1]$ com uma distribuição uniforme. Esta estratégia permite que a nova solução se aproxime da mesma região de espaço de busca de \vec{M} , como também, se afaste dessa região. O deslocamento das novas soluções no espaço de busca dependem da diversidade da população. Portanto, com uma população sem diversidade, a relação de comensalismo não realiza perturbações na solução \vec{X}_i .

A relação de parasitismo utiliza uma solução candidata (\vec{X}_i) como um parasita. Esta solução é duplicada e uma dimensão k é selecionada aleatoriamente para ser modificada. O novo valor de P_k é determinado aleatoriamente dentro do domínio do problema otimizado (LB e UB). O parasita se associa com a solução \vec{X}_j e tenta substituí-lo na população. Como pode-se observar, a geração de novas soluções na relação de parasitismo não é influenciada por outras soluções e todas as decisões necessárias são tomadas aleatoriamente. A relação de parasitismo, ao contrário das relações de mutualismo e comensalismo, não é influenciada pela diversidade da população. A relação funciona da mesma maneira em uma população com um baixo ou alto nível de diversidade. Além

Tabela I. FUNÇÕES MATEMÁTICAS

Função	Domínio	Min.
1 - Ackley	$[-32, 32]^d$	0
2 - Griewank	$[-600, 600]^d$	0
3 - Levy	$[-10, 10]^d$	0
4 - Rastrigin	$[-5, 12; 5, 12]^d$	0
5 - Rosenbrock	$[-30, 30]^d$	0
6 - Schaffer F7	$[-100, 100]^d$	0
7 - Schwefel 2.22	$[-10, 10]^d$	0
8 - Shifted Ackley	$[-32, 32]^d$	-140
9 - Shifted Griewank	$[-600, 600]^d$	-180
10 - Shifted Rastrigin	$[-5, 12; 5, 12]^d$	-330
11 - Shifted Rosenbrock	$[-100, 100]^d$	390
12 - Shifted Schaffer	$[-100, 100]^d$	0
13 - Shifted Schwefel Problem 2.21	$[-100, 100]^d$	-450
14 - Shifted Sphere	$[-100, 100]^d$	-450
15 - Sphere	$[-100, 100]^d$	0
16 - Step	$[-100, 100]^d$	0
17 - Zakharov	$[-5, 10]^d$	0

disso, esta relação permite que a nova solução se desloque no espaço de busca de maneira que não é possível com as relações de mutualismo e comensalismo. O deslocamento que a relação de parasitismo realiza é aleatório e permite que a solução salte para qualquer ponto no espaço de busca. Nas relações de mutualismo e comensalismo, o deslocamento depende das soluções envolvidas.

A cada iteração do algoritmo, são aplicadas as três relações simbióticas em cada solução candidata. Portanto, a cada solução, são aplicados dois componentes de intensificação e apenas um de diversificação. Considerando a quantidade de avaliações da função objetivo, são realizadas três avaliações da função através dos componentes de intensificação e uma avaliação com o componente de diversificação, lembrando que a relação de mutualismo realiza duas avaliações da função objetivo. Além disso, quando a população perde a diversidade, as relações de mutualismo e comensalismo não são capazes de gerar novas soluções. A partir dessa análise, existe uma percepção de que a intensificação e a diversificação podem estar desequilibradas no algoritmo.

A análise da performance do SOS com todas as combinações de relações simbióticas permite a avaliação da real contribuição dos componentes durante o processo de otimização e é uma maneira de se verificar se o algoritmo possui ou não um desequilíbrio entre a intensificação e a diversificação. A combinação que obter a melhor performance, consequentemente, representa o melhor equilíbrio entre a intensificação e a diversificação. As possíveis combinações de relações simbióticas são: o algoritmo canônico SOS, SOS_{mc} com o mutualismo e comensalismo, SOS_{mp} com o mutualismo e parasitismo, SOS_{cp} com o comensalismo e parasitismo, SOS_m com o mutualismo, SOS_c com o comensalismo e finalmente o SOS_p com o parasitismo. Os experimentos realizados são descritos na seção seguinte.

IV. EXPERIMENTOS E RESULTADOS

Os algoritmos foram desenvolvidos utilizando a linguagem de programação C++ e os experimentos foram executados sobre AMD Phenon II X4 (2.80GHz) com 4GB RAM, com sistema operacional Linux. Os experimentos foram conduzidos utilizando 17 funções contínuas utilizadas extensivamente na literatura [11]. A Tabela I apresenta o nome da função, o domínio e o ponto mínimo de cada função. Todas as funções caracterizam problemas de minimização sem restrições.

Os algoritmos usados nos experimentos são SOS, SOS_{cp} , SOS_{mc} , SOS_{mp} , SOS_c , SOS_p e SOS_m . Para cada função de teste são utilizadas duas dimensionalidades, 30 e 200 dimensões, para verificar o desempenho dos algoritmos com diferentes graus de complexidade. São realizadas 30 execuções independentes com populações de 50 organismos inicializados aleatoriamente. A condição de parada usada é 100 000 avaliações da função objetivo e o ponto ótimo é considerado com a diferença mínima de 10^{-20} .

Os resultados dos experimentos são apresentados na Tabela II, para 30 dimensões, e na Tabela III, para 200 dimensões. As tabelas mostram o número da função, a média e o desvio padrão dos resultados obtidos para os algoritmos SOS, SOS_{cp} , SOS_{mc} , SOS_{mp} , SOS_c , SOS_p e SOS_m , respectivamente. O valor em negrito de cada linha corresponde ao algoritmo com

melhor resultado dentre os demais, levando em consideração a análise estatística. Na parte inferior da tabela é contabilizada a quantidade de melhores soluções de cada algoritmo e o ranqueamento feito pelo *Friedman Rank Sum Test*, com valor em negrito correspondente a melhor posição no ranque.

A análise estatística foi feita através de dois testes não-paramétricos: *Friedman Rank Sum Test* e *Wilcoxon Signed Rank Test* [12]. Optou-se por testes não-paramétricos dada a não-normalidade das variáveis envolvidas verificada com a aplicação do teste *Shapiro-Wilk* ($p\text{-value}=0,05$) [13]. Ao aplicar o teste de *Friedman* é possível detectar se há diferenças significativas entre os valores obtidos pelos algoritmos e determinar se o valor das amostras selecionadas ao acaso são realmente aleatórias e diferentes. Por este teste, quanto menor o ranqueamento obtido melhor o resultado. O *Friedman Test* faz a análise entre todos os algoritmos e não é possível tirar conclusões mais precisas de comparação pareada, ou seja, comparar o melhor entre cada par de algoritmos. Por isso o teste *Wilcoxon* é utilizado como complemento.

O teste *Wilcoxon* compara os resultados de dois algoritmos para identificar se existe diferenças com significância estatística entre eles. Para analisar estatisticamente que um algoritmo é melhor do que outro, é verificada a hipótese nula em que os resultados são equivalentes e espera-se rejeitar esta hipótese. Rejeitando a hipótese nula, tem-se que um algoritmo é melhor que outro. O teste de *Wilcoxon* é aplicado com hipótese nula de $p\text{-value}=0,05$ para cada função (30 e 200 dimensões) e pareia-se os resultados dos algoritmos SOS, SOS_{mc} , SOS_{mp} , SOS_c , SOS_p e SOS_m com o algoritmo SOS_{cp} . O pareamento foi realizado com o SOS_{cp} pois os resultados preliminares se mostraram mais promissores para esta abordagem.

A Tabela II apresenta os resultados obtidos para as 17 funções de testes com 30 dimensões. Na tabela, o algoritmo SOS_{cp} obteve melhor resultado em 15 das 17 funções, exceto em $f4$ e $f13$. O algoritmo SOS_p apresentou melhor resultado em $f4$. Na função $f15$ os algoritmos SOS e SOS_{cp} obtiveram resultados estatisticamente equivalentes e são melhores em comparação aos demais algoritmos. O teste *Wilcoxon* obteve $p\text{-value}$ maior que 0.05 apenas no pareamento SOS_{cp} e SOS na função $f13$, confirmando a hipótese nula de que os valores são

estatisticamente equivalentes. Em todas as demais funções o p -value foi menor que 0,05, rejeitando a hipótese nula e indicando que os resultados nos experimentos são estatisticamente diferentes. Os demais algoritmos SOS_{mc} , SOS_{mp} , SOS_c , e SOS_m não foram estatisticamente melhores em nenhum caso. Os valores obtidos para o *Friedman Test* estão na parte inferior da Tabela II e a abordagem com comensalismo e parasitismo, SOS_{cp} , foi a de melhor *ranking*.

Tabela II. RESULTADOS DOS EXPERIMENTOS COM ($d = 30$)

f	SOS	SOS_{cp}	SOS_{mc}	SOS_{mp}
1	0,69±0,28	0,34±0,08	12,59±6,21	4,15±0,36
2	0,012±0,00	0,008±0,002	0,03±0,01	1,44±0,29
3	0,05±0,01	0,004±0,001	28,37±10,48	0,49±0,10
4	18,51±2,27	6,59±1,69	273,86±18,66	18,49±1,73
5	322,87±95,34	85,67±45,36	16684,1±16174,6	3677,99±1514,66
6	6,62±2,07	4,37±1,91	11,29±0,62	14,23±1,26
7	0,19±0,03	0,27±0,05	0,83±1,77	3,93±0,39
8	-138,32±0,37	-139,65±0,07	-132,18±1,45	-135,75±0,28
9	-178,91±0,019	-179,12±0,07	-171,44±7,17	-177,67±0,23
10	-317,34±2,94	-329,35±0,55	-129,43±13,02	-314,8±2,15
11	9431,59±4794,82	1046,24±354,56	1,93e+08±3,04e+08	235765±80
12	36,26±4,38	21,49±2,74	191,61±11,86	59,74±4,83
13	-444,80±0,87	-447,66±0,42	-420,65±5,98	-439,3±1,20
14	-442,17±2,35	-449,19±0,19	957,91±2230,27	-311,61±30,05
15	1,04±0,42	0,87±0,21	4,08±1,55	143,55±32,73
16	3,55±0,69	0,86±0,33	10,61±2,97	139,68±27,56
17	150,77±61,86	47,15±34,52	389,37±62,79	454,87±161,82
Qtd. Melhores	1	15	0	4
Friedman Rank	2 ^a	1 ^a	5 ^a	4 ^a

f	SOS_c	SOS_p	SOS_m
1	16,26±1,90	1,96±0,19	19,91±0,12
2	1,138±0,29	0,16±0,04	409,06±63,51
3	43,18±9,24	0,07±0,01	141,87±28,39
4	284,20±13,34	4,41±0,74	371,03±22,83
5	68469,9±34900,5	373,32±94,71	1,26e+08±2,67e+07
6	11,81±0,51	15,94±0,78	15,76±0,59
7	9,45±1,23	1,42±0,16	94,10±7,93
8	-130,02±0,85	-138,16±0,25	-119,25±0,32
9	-178,86±0,03	-178,84±0,025	476,91±130,26
10	-106,05±13,67	-328,37±0,41	117,59±36,92
11	752136±219005	7802,96±2838,51	4e+10±1,51e+10
12	202,15±15,02	33,78±2,57	287,12±10,84
13	-420,20±2,94	-446,053±0,62	-343,82±7,74
14	-441,34±2,002	-431,059±3,52	70884,6±16142,3
15	104,04±32,24	17,48±3,64	45788,9±5108,05
16	91,42±17,90	17,35±3,06	47231,2±4821,94
17	238,73±34,94	251,95±60,45	1214,39±219,24
Qtd. Melhores	0	1	0
Friedman Rank	6 ^a	3 ^a	7 ^a

Na Figura 1 é apresentada a convergência da otimização na função Rosenbrock. Através do gráfico é possível observar que o SOS_{cp} obteve a melhor convergência. Isto indica que o SOS_{cp} é capaz de identificar rapidamente as áreas promissoras do espaço de busca e explorá-las por mais tempo. A maioria das abordagens, com exceção do SOS_m poderiam continuar o processo de otimização, devido não estarem presos em um ponto local. SOS_m estagnou prematuramente em um ponto local devido esta abordagem não possuir quaisquer mecanismo de diversificação para escapar do ponto local. Este padrão de comportamento foi observado nas demais funções.

A Tabela III mostra os resultados obtidos para as 17 funções de testes com 200 dimensões. Na tabela, o algoritmo SOS_{cp} obteve melhor resultado em 12 das 17 funções, exceto em f_4, f_6, f_{10}, f_{13} e f_{17} . O algoritmo SOS_p obteve melhor resultado em f_4, f_{10}, f_{13} e f_{17} . O teste *Wilcoxon* obteve p -value maior que 0,05 apenas no pareamento SOS_{cp} e SOS_{mp} na função f_6 , confirmando a hipótese nula de que os valores são estatisticamente equivalentes. Nas demais funções o p -value foi menor que 0,05, rejeitando a hipótese nula e indicando que os resultados nos experimentos são estatisticamente diferentes. Os algoritmos SOS , SOS_{mc} , SOS_{mp} , SOS_c , e SOS_m não foram estatisticamente melhores em nenhum caso. Os valores para o *Friedman Test* estão na parte inferior da Tabela III e, novamente, a abordagem com comensalismo e parasitismo, SOS_{cp} , foi a de melhor *ranking*.

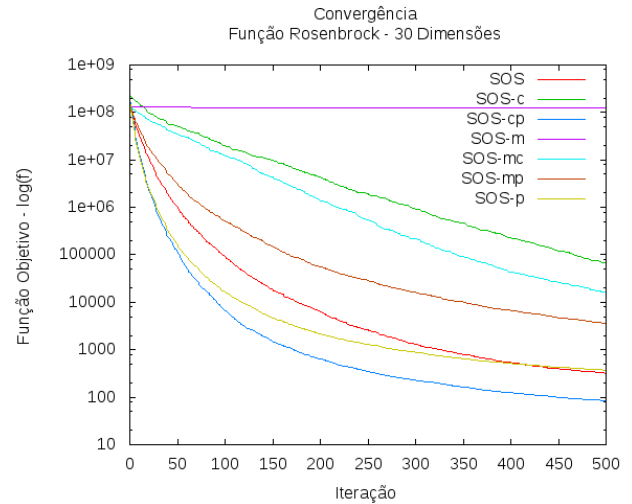


Figura 1. Convergência para a função Rosenbrock com 30 dimensões

Tabela III. RESULTADOS DOS EXPERIMENTOS COM ($d = 200$)

f	SOS	SOS_{cp}	SOS_{mc}	SOS_{mp}
1	19,87±0,01	6,00±0,25	19,96±0,00	19,29±1,65
2	99,27±11,55	24,63±2,52	3992,24±214,00	291,04±20,90
3	44,71±5,29	11,66±1,11	1850,35±91,08	75,94±7,22
4	610,32±36,55	376,51±22,19	3037,12±59,99	724,29±34,12
5	41e+10±58e+5	40e+4±10e+4	2,03e+12±1,54e+08	1,72e+10±1,92e+9
6	27,21±0,53	27,82±0,46	27,28±0,45	27,17±0,54
7	74,72±4,48	33,34±2,22	733,05±17,31	171,56±7,06
8	-129,16±0,57	-133,68±0,26	-118,71±0,07	-126,19±0,41
9	-62,40±16,27	-153,90±2,74	2678,13±324,49	191,91±27,83
10	362,61±32,78	77,22±21,87	2485,64±107,15	507,49±39,08
11	7,23e+08±1,18e+08	4,87e+07±1,11e+07	1,64e+11±2,72e+10	3,38e+09±5,35e+08
12	906,22±50,99	628,70±28,20	2312,69±32,44	1055,65±31,98
13	-359,52±4,45	-383,17±3,74	-299,95±3,28	-359,18±5,67
14	12203,7±1254,15	2367,31±293,05	301838±42084,8	41808,8±3248,73
15	11023,3±1117,09	2614±280,66	444932±20745,4	31844,4±2921,33
16	11255,2±1035,6	2624,14±260,11	443349±18403,9	31216,1±2477,23
17	4747,06±558,71	4253,62±267,01	6110,09±421,24	3898,27±1081,34
Qtd. Melhores	0	13	0	1
Friedman Rank	3 ^a	1 ^a	5 ^a	4 ^a

f	SOS_c	SOS_p	SOS_m
1	20,83±0,01	7,16±0,14	19,96±0,00
2	4692,54±281,26	46,21±3,37	4337,77±198,15
3	2102,35±95,25	12,71±0,74	1961,57±75,56
4	3202,24±68,57	282,32±10,00	3102,58±57,87
5	2,57e+09±1,29e+08	612953±99336,1	2,15e+09±1,46e+08
6	28,15±0,50	27,94±0,53	27,40±0,50
7	793,77±19,41	63,97±2,54	809,06±16,06
8	-118,65±0,01	-132,64±0,15	-118,62±0,025
9	4026,76±245,90	-128,37±4,16	8307,46±387,95
10	3151,32±121,69	-26,95±11,98	4194,13±166,67
11	4,11e+11±5,64e+10	7,38e+07±1,01e+07	9,91e+11±6,57e+107
12	2359,09±39,08	267,94±24,15	2370,14±39,45
13	-295,94±2,95	-401,05±2,76	-286,97±3,91
14	462654±33137,2	5390,93±404,42	1,01e+06±49202,37
15	524354±24201,3	5050,62±356,95	487241±18719,57
16	523838±21422,9	5185,44±350,52	484330±19839
17	4864,66±194,79	3078,55±232,13	3,86e+16±3,19e+16
Qtd. Melhores	0	4	0
Friedman Rank	6 ^a	2 ^a	7 ^a

Na Figura 2 é apresentada a convergência da otimização na função Rosenbrock. Através do gráfico é possível observar que o SOS_{cp} foi a primeira abordagem a convergir e obteve a melhor convergência. Isto indica que o SOS_{cp} é capaz de identificar rapidamente as áreas promissoras do espaço de busca e explorá-las por mais tempo. SOS_p apresentou uma convergência semelhante ao SOS_{cp} , porém obteve um resultado inferior. A boa performance do SOS_p indica que a diversificação é extremamente importante no processo de otimização. Apesar de ser uma abordagem apenas com uma rotina de diversificação, obteve resultados superiores que a maioria das abordagens, com exceção de SOS_{cp} . As estratégias SOS_c , SOS_m e SOS_{mc} , que possuem apenas rotinas de intensificação estagnaram rapidamente em um ponto local. Já as demais abordagens, SOS_{cp} , SOS_p , SOS_{mp} e SOS poderiam

continuar o processo de otimização, devido não estarem presos em um ponto local. Este padrão de comportamento foi observado nas demais funções.

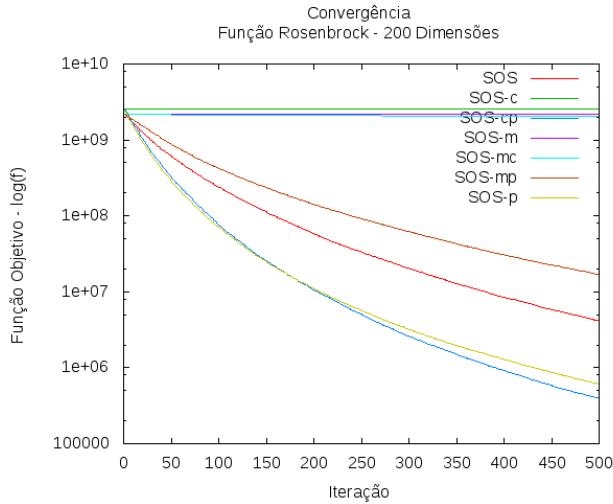


Figura 2. Convergência para a função Rosenbrock com 200 dimensões

A. Discussão

No algoritmo SOS canônico, as relações de mutualismo e comensalismo são responsáveis pela intensificação e a relação de parasitismo é responsável pela diversificação. Na rotina de mutualismo, o processo de intensificação aproxima as demais soluções da melhor solução da população. Já no comensalismo, de maneira estocástica, a intensificação tem a possibilidade de aproximar ou se distanciar da melhor solução da população. É importante notar que ambos se tornam inoperantes quando a diversidade da população é baixa, pois com soluções próximas, a aproximação ou distanciamento é mínimo ou nulo. O parasitismo não depende da diversidade atual da população e gera novas soluções de modo estocástico, permitindo manter diversidade na população.

Ao excluir a rotina de mutualismo do algoritmo SOS percebe-se um melhor equilíbrio entre diversificação e intensificação, conforme os resultados obtidos pela abordagem SOS_{cp} . O comensalismo se mostrou uma estratégia melhor de intensificação, dada a característica estocástica no direcionamento da população. Combinado com a manutenção da diversidade gerada pelo parasitismo, o equilíbrio entre diversificação e intensificação se mostraram promissores com apenas estes dois componentes simbióticos. Pode-se perceber pelos resultados obtidos que, em todas as abordagens que utilizam a relação de mutualismo, os resultados obtidos não são promissores. Isto indica que a rotina de mutualismo favorece de maneira excessiva a intensificação do processo de busca. O alto nível de intensificação pode ser explicado pelo fato deste componente aplicar duas avaliações da função e ter como objetivo explorar apenas a região da melhor solução da população.

V. CONCLUSÃO

O equilíbrio adequado entre a diversificação e intensificação permite explorar com mais eficiência o espaço

de busca, e consequentemente, obter resultados melhores. O SOS é composto por três componentes simbióticos: mutualismo e comensalismo, responsáveis pela intensificação, e o parasitismo, responsável pela diversificação da busca. A análise dos resultados mostrou que a relação de mutualismo apresenta um alto nível de intensificação, e consequentemente, prejudica o processo de otimização. Experimentos combinando todas as possibilidades de combinação dos componentes simbióticos indicaram que ao utilizar o SOS sem a relação de mutualismo, foi possível obter um ganho de performance significativo. A utilização das relações de comensalismo e parasitismo, abordagem SOS_{cp} , permitiu a identificação de áreas mais promissoras no espaço de busca.

Como trabalhos futuros pretende-se realizar a análise do desempenho do SOS_{cp} em relação com outros algoritmos do estado da arte, a aplicação do SOS_{cp} em problemas reais, a utilização assíncrona dos componentes do algoritmo e o desenvolvimento de um mecanismo para anular a falta de perturbação na relação de comensalismo quando a população está com pouca ou sem diversidade.

Agradecimentos: Os autores gostariam de agradecer à Fundação de Amparo a Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) pelo apoio financeiro, como também a Universidade do Estado de Santa Catarina (UDESC).

REFERÊNCIAS

- [1] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," *Int. J. Bio-Inspired Comput.*, vol. 3, no. 1, pp. 1–16, Feb. 2011.
- [2] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer, 2006.
- [3] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 36–39.
- [4] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm," *Computers & Structures*, vol. 139, pp. 98–112, 2014.
- [5] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.
- [6] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.
- [7] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.
- [8] M. Aulady, "A hybrid symbiotic organisms search-quantum neural network for predicting high performance concrete compressive strength," 2014.
- [9] M.-Y. Cheng, D. Prayogo, and D.-H. Tran, "Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search," *Journal of Computing in Civil Engineering*, 2015.
- [10] K. Narasimhamu, V. V. Reddy, and C. Rao, "Optimization of buffer allocation in tandem closed queueing network using symbiotic organisms search," *International Journal of Applied Engineering Research*, vol. 10, no. 5, 2015.
- [11] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.
- [12] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability," *Soft Computing*, vol. 13, no. 10, pp. 959–977, 2009.
- [13] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, pp. 591–611, 1965.