3
Abordagem proposta para calibragem de alinhadores

Dadas duas ontologias, deseja-se encontrar o grau de similaridade entre entidades de cada ontologia aplicando, para tal, um conjunto de funções de similaridade. Embora abordagens para identificar relevância de funções de similaridade sejam conhecidas, como é o caso do sistema RiMOM (Li et al, 2009), as heurísticas utilizadas por estas abordagens são dependentes das funções de similaridade utilizadas, o que impede a inserção de novas funções de similaridade na heurística. Além disso, algumas abordagens necessitam realizar um pré-processamento das ontologias, o que pode influenciar no desempenho do sistema.

Porém, em muitos casos, o engenheiro de ontologias pode determinar facilmente similaridades conhecidas (geralmente pares de conceitos muito similares ou pouco similares). Esta informação pode ser utilizada para descobrir qual a melhor configuração de pesos para um conjunto de funções de similaridade a serem aplicadas em duas ontologias. A abordagem descrita neste trabalho parte do princípio de que engenheiros de ontologias podem fornecer exemplos de correspondências entre entidades para, em seguida, calibrar alinhadores automaticamente.

Assim, o GNoSIS+ é um algoritmo genético elitista que trabalha com o paradigma de estratégia uni-objetivo. Um algoritmo genético é chamado de elitista quando este garante que as melhores soluções se preservem ao longo das gerações. Um algoritmo genético uni-objetivo é aquele que possui uma única função objetivo (Costa, 2003). O GNoSIS+ utiliza uma abordagem para calibrar automaticamente uma composição de alinhadores a partir de exemplos de correspondências fornecidos por engenheiros de ontologias, a qual objetiva-se ser uma abordagem que privilegie o desempenho do sistema sem detrimento da qualidade da solução.

Este capítulo apresenta a heurística proposta para solução do problema de calibragem em meta-alinhadores de ontologias. A seção 3.1 apresenta como o problema foi modelado, enquanto a seção 3.2 aborda como cada solução é representada. A heurística proposta é formada pelos operadores de construção

das soluções (seção 3.3), de reprodução e mutação (seção 3.4), de intensificação da solução (seção 3.5) e de construção de gerações (seção 3.6). Por fim, a seção 3.7 apresenta o comportamento da heurística em diferentes configurações para os operadores genéticos.

3.1 Descrição do problema

Considere uma função de similaridade composta f e um conjunto S de correspondências de equivalência conhecidas. O conjunto S é formado por tuplas $(x_i, y_i, =, s_i)$, onde x_i e y_i são entidades de ontologias distintas, $=$ denota a relação do tipo equivalência e s_i é a similaridade conhecida entre x_i e y_j. Ao aplicar a função f em x_i e y_i, espera-se encontrar o valor s_i, ou seja, $f(x_i, y_i) = s_i$. Como exemplo, considere o conjunto $S' = \{(x_1, y_1, =, 1), (x_2, y_2, =, 1), (x_3, y_3, =, 1)\}$. Neste conjunto, todas as correspondências conhecidas possuem grau de similaridade igual a 1. Considerando uma função $\bar{f}'(x, y) = g_1(x, y)p_1 + g_2(x, y)p_2 + g_3(x, y)p_3$, tem-se que:

$$\begin{aligned} \bar{f}'(x_1, y_1) = s_1 &\therefore g_1(x_1, y_1)p_1 + g_2(x_1, y_1)p_2 + g_3(x_1, y_1)p_3 = 1 \\ \bar{f}'(x_2, y_2) = s_2 &\therefore g_1(x_2, y_2)p_1 + g_2(x_2, y_2)p_2 + g_3(x_2, y_2)p_3 = 1 \\ \bar{f}'(x_3, y_3) = s_3 &\therefore g_1(x_3, y_3)p_1 + g_2(x_3, y_3)p_2 + g_3(x_3, y_3)p_3 = 1 \end{aligned} \tag{3-1}$$

No sistema de equações acima, as funções g_i são constantes do sistema e deseja-se encontrar os pesos p_i que melhor resolvam essa equação. Como este pode ser um sistema impossível, deseja-se encontrar os pesos p_i que forneçam a melhor aproximação de s_i, ou seja, dado um conjunto de ocorrências conhecidas S e uma função de similaridade composta f, deseja-se determinar os pesos p_i que, aplicados aos membros g_i de f fazem com que f seja normalizada de tal forma que:

$$\sum_{i=1}^{|S|} s_i - \sum_{i=1}^{|S|} f(x_i, y_i) \approx 0 \tag{3-2}$$

Uma vez definidos os pesos p_i para os membros de f, a função de similaridade f poderá ser aplicada para calcular a similaridade das demais entidades das duas ontologias.

3.2 Representação da solução

Considera-se que funções de similaridade simples podem ser agregadas em similaridades compostas (definição 2.3). Estas funções de similaridades compostas, por sua vez, podem ser utilizadas em outras composições. É gerada,

assim, uma árvore de composição de funções, denotada aqui pela letra Υ. Cada composição será normalizada através da aplicação de pesos conforme a definição 2.5. Assim, cada nó interno de Υ possuirá um peso associado ao valor de retorno da função em relação aos dois elementos de entrada e e e'. A raiz de Υ, por sua vez, armazenará o valor final da equação representada pela árvore de composição.

Exemplo 3.1 *Considere a composição abaixo, onde F_1, F_2, F_4 e F_5 são funções não compostas e P_x são pesos.*

$$\begin{aligned} F_3 &= F_1P_4 + F_2P_5 \\ F_3P_1 + F_4P_2 + F_5P_3 &= K_1 \end{aligned} \tag{3-3}$$

A equação pode ser reduzida para forma:

$$(F_1P_4 + F_2P_5)P_1 + F_4P_2 + F_5P_3 = K_1 \tag{3-4}$$

Esta equação gera uma árvore como a da figura 3.1.

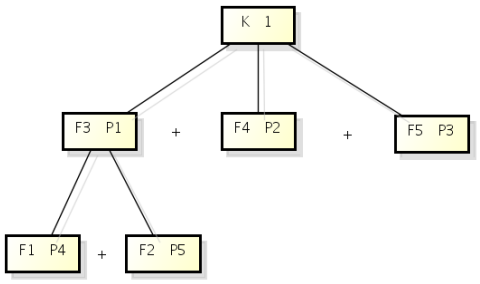

Figura 3.1: Representação da equação $(F_1P_4 + F_2P_5)P_1 + F_4P_2 + F_5P_3 = K_1$

Na abordagem apresentada neste trabalho, é utilizada uma representação com números reais na qual um indivíduo (cromossomo) é representado por um conjunto C de genes g de forma que $C = \{g_1, g_2, g_3, \ldots, g_n\}$ com $g_i \in [0, 1]$. Assim, uma população S é definida como sendo um conjunto de cromossomos C, ou seja, $S = \{C_1, C_2, C_3, \ldots, C_n\}$. O tamanho da população é dado por $|S|$.

Cada gene g representa uma variável do sistema de equações, de tal forma que, dada uma sequência de funções Ξ formada pelo percurso em pós-ordem em Υ, com exceção do nó raiz, o peso g_i representa o peso a ser aplicado ao i-ésimo elemento de Ξ. Cada cromossomo possui um número de genes igual ao tamanho de Ξ.

Exemplo 3.2 *O percurso em pós-ordem na árvore de similaridade composta apresentada na figura 3.1 gera uma sequência* $\Xi = \{F_1, F_2, F_3, F_4, F_5\}$. *Para encontrar uma solução que representa a melhor calibragem para essas funções, são criados cromossomos contendo 5 genes (*$|\Xi| = 5$*) na forma* $C = \{g_1, g_2, g_3, g_4, g_5\}$, *onde* g_i *é um número real pertencente a* $[0, 1]$ *que representa o peso a ser aplicado à função* $F_i, 1 \leq i \leq 5$.

Uma sequência $\Xi^{e,e'}$ representa a sequência de valores $v_1, v_2, ..., v_n$ de similaridades entre os elementos e e e', onde v_i é igual a $F_i(e, e')$ e $F_i \in \Xi$.

Considerando uma correspondência conhecida entre dois elementos e e e' quaisquer de ontologias distintas e um cromossomo C, é possível utilizar uma função $\alpha : \Xi^{e,e'} \times C \mapsto \Re$ que retorna o valor da similaridade entre os elementos e e e' após a aplicação de cada peso $g_i \in C$ nos valores de similaridade $v_i \in \Xi^{e,e'}$.

Exemplo 3.3 *Seja uma sequência* $\Xi^{e,e'} = \{F_1(e, e'), F_2(e, e'), F_3(e, e')\}$. *Para tornar o exemplo mais simples, considere que* $\Xi^{e,e'}$ *é formada somente por funções simples, ou seja, a árvore de funções compostas possui altura 2. Dado um cromossomo* $C = \{g_1, g_2, g_3\}$, *a função* α *calcula a aplicação dos pesos contidos no cromossomo* C *aos valores contidos em* $\Xi^{e,e'}$. *Assim, para esse exemplo,* $\alpha(\Xi^{e,e'}, C) = F_1(e, e')g_1 + F_2(e, e')g_2 + F_3(e, e')g_3$.

O GNoSIS+ parte do princípio de que algumas correspondências podem ser facilmente identificadas por engenheiros de ontologias sem muito esforço, mesmo em ontologias muito grandes. Assim, seja CP as correspondências fornecidas por engenheiros de ontologias, o GNoSIS+ trata o problema como uma minimização da inequação 3-5, onde s é o grau de similaridade definido pelo engenheiro de ontologias para a correspondência entre e e e'.

$$\sum_{i=1}^{|CP|} |s_i - \alpha(\Xi^{e_i,e'_i}, C)| \geq 0 \tag{3-5}$$

A equação 3-5 é denominada *função de aptidão* ou *função de avaliação*, denotada por $\vartheta : \Xi \times C \mapsto \Re$. Esta função é utilizada para verificar quanto uma solução (cromossomo) se aproxima da solução final do problema. Quanto mais próximo de zero for o valor da função de aptidão, mais a solução se aproxima da solução final.

Exemplo 3.4 *Considere* $CP = \{(e_1, e'_1, =, 1), (e_2, e'_2, =, 1)\}$ *e o valor das funções* $\alpha(\Xi^{e_1,e'_1}, C) = 1.05$ *e* $\alpha(\Xi^{e_2,e'_2}, C) = 0.9$, *o valor da função de aptidão para o cromossomo* C *é calculado como* $\vartheta(\Xi, C) = |1 - 1.05| + |1 - 0.9| = 0.15$.

Uma função de aptidão que retorna o quão próximo uma solução armazenada no cromossomo C está do somatório das correspondências fornecidas pelos especialistas é denominada ϑ^- e é definida como abaixo:

$$\vartheta^-(\Xi, C) = (\sum_{i=1}^{|CP|} s_i) - \vartheta(\Xi, C) \tag{3-6}$$

Exemplo 3.5 *Considerando as variáveis definidas no exemplo 3.4, o valor da função de aptidão ϑ^- para o cromossomo C é calculado como $\vartheta^-(\Xi, C) = (1+1) - (1.05 + 0.9) = 0.5$.*

A função de aptidão ϑ representa o objetivo que a heurística visa atingir, ou seja, encontrar uma solução que faça com que ϑ retorne o valor mais próximo de 0. Contudo, uma segunda função de aptidão ϑ^- é utilizada em certos operadores genéticos, conforme será visto nas seções seguintes.

3.3 Construtor das soluções iniciais

O problema descrito na seção 3.1 possui infinitas soluções. Para reduzir o espaço de possíveis soluções, discretizamos o problema ao introduzir uma granularidade τ onde todos os pesos g são múltiplos de τ. O conjunto de múltiplos da granularidade τ é denotado por $\dot{\tau}$, onde o menor elemento de $\dot{\tau}$ é zero e o maior elemento é igual a 1.

Uma população inicial com um número determinado de indivíduos é construída de forma aleatória. A criação de um cromossomo C se dá com o sorteio de um gene do cromossomo e, em seguida, é sorteado um peso $g_i \in \dot{\tau}$ de tal forma que $g_i \leq 1 - \sum_{i=1}^{|\check{C}|} \check{g}_i$, onde $\check{C}$ denota os múltiplos de $\dot{\tau}$ já sorteados para o cromossomo C. Assim, a soma dos pesos de uma solução inicial é sempre menor ou igual a 1.

Vale salientar que o processo de construção é feito de forma recursiva, uma vez que os genes do cromossomo representam pesos aplicados às funções de similaridade em diferentes níveis na árvore Υ. Então, o processo é iniciado para cada conjunto de genes que estão em um mesmo nível em Υ. O algoritmo 2 apresenta o processo de construção das soluções iniciais.

Algoritmo 2: Criação da população inicial

Entrada: $g \leftarrow$ granularidade; $T \leftarrow$ total de indivíduos
Saída: Pop (População inicial)
início
  enquanto $|Pop| \leq T$ **faça**
    $C_i \leftarrow$ Novo cromossomo
    para cada $g_j \in C_i \wedge g_j \notin \check{C}$, *de forma aleatória* **faça**
      $p \leftarrow$ posição aleatória de $\dot{\tau}$, tal que $1 \leq p \leq (|\dot{\tau}|$ - soma das posições já sorteadas de $\dot{\tau})$.
      $g_j \leftarrow \dot{\tau}[p]$
      $C_i \leftarrow C_i \cup g_j$
    fim para cada
    $Pop \leftarrow Pop \cup C_i$
  fim enquanto
fim

Exemplo 3.6 *Este exemplo demonstra o processo de construção de uma solução inicial com base na equação 3-4. É criado com um cromosso, como o da figura 3.2, onde as duas primeiras posições representam os pesos para as funções de segundo nível* (N_2) *e as últimas três posições os pesos para as funções de primeiro nível* (N_1) *(ver árvore da figura 3.1).*

N_2	N_2	N_1	N_1	N_1

Figura 3.2: Cromossomo com destaque para a posicão de cada nível

Considerando $\tau = 0.05$*, primeiro são sorteados valores aleatórios para cada gene denotado por* N_1*, em ordem aleatória, conforme ilustrado na figura 3.3.*

N_2	N_2	0.35	0.50	0.15

Figura 3.3: Exemplo de cromossomo com valores válidos para funções do primeiro nível

Em seguida, o processo é repetido para os genes denotados por N_2*, podendo gerar um cromossomo com os valores representados na figura 3.4.*

0.65	**0.30**	0.35	0.50	0.15

Figura 3.4: Exemplo de um cromossomo com valores válidos

Note que a soma dos valores é sempre menor ou igual 1 para cada nível da árvore Υ da figura 3.1.

3.4 Operadores de reprodução e mutação

Cruzamentos entre indivíduos ocorrem a uma taxa especificada t_c em relação ao tamanho da população. Para realizar a seleção de cada par de indivíduo para cruzamento, é utilizada uma técnica de seleção natural conhecida algoritmo da roleta (*Roulette Wheel*) (Goldberg, 1989), uma vez que, por este método, a probabilidade de seleção de um cromossomo é diretamente proporcional à sua aptidão. Tal característica permite que a carga genética dos indivíduos mais aptos tenha maior probabilidade de ser compartilhada nas novas gerações. Seja f_i a aptidão do indivíduo p_i na população P e n o tamanho da população P, a probabilidade do indivíduo p_i ser selecionado é:

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{3-7}$$

Esta técnica consiste em realizar uma ordenação descendente por aptidão dos invidíduos da população S e, então, é sorteado um indivíduo com base na sua posição no conjunto. Assim, indivíduos com uma maior aptidão terão uma maior "fatia" da roleta e, com isso, uma maior probabilidade de serem escolhidos.

Para este trabalho, contudo, a função de avaliação utilizada para verificar a aptidão do indivíduo (ver equação 3-5) é inversamente proporcional à qualidade da solução. Ou seja, o indivíduo que representa uma solução de melhor qualidade possui um valor menor na sua função de aptidão. Com isso, para que a técnica da roleta possa ser aplicada, é utilizada na ordenação dos indivíduos a função de aptidão complementar descrita na equação 3-6. Essa função de avaliação retorna um valor diretamente proporcional à qualidade da solução que o indivíduo representa. Assim, indivíduos que armazenam melhores soluções recebem um maior valor numérico e terão maior chance de serem escolhidos.

Uma vez ordenados os indivíduos da população, é sorteado um número real randômico $r \in [0, \theta_S^-]$, onde θ_S^- é a aptidão complementar total da população S (equação 3-8).

$$\theta_S^- = \sum_{i=1}^{|S|} \vartheta^-(\Xi, C_i) \tag{3-8}$$

Em seguida, a população S é percorrida sequencialmente e é escolhido o cromossomo C_p, na posição p, tal que

$$min(\sum_{i=1}^{p} \vartheta^{-}(\Xi, C_p)) \geq r \tag{3-9}$$

A figura 3.5 ilustra como o algorítmo funciona.

N	Cromossomo	Fitness	Graus
1	0001100101010	6.0	180
2	0101001010101	3.0	90
3	1011110100101	1.5	45
4	1010010101001	1.5	45

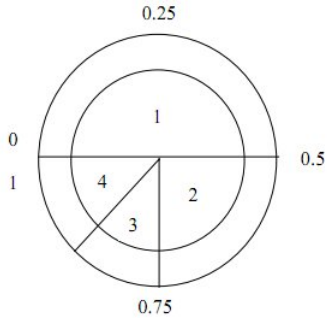

Figura 3.5: Técnica da roleta para seleção de indivíduos (Gudwin & von Zuben, 1998)

Exemplo 3.7 *Supondo a escolha de um número aleatório $r = 0.65$ e a população da figura 3.5, o cromossomo escolhido será o cromossomo 2, uma vez que, ao percorrer a população de forma sequencial, o primeiro cromossomo possui aptidão igual 0.5, o segundo cromossomo possui aptidão igual a 0.25 e a soma das aptidões do cromossomo 1 e 2 fornece a soma da menor sequência de aptidões maior que o r, ou seja, 0.25+0.5 $\geq$ 0.65.*

Dados dois indivíduos $C_1 = (g_1, g_2, \ldots, g_l)$ e $C_2 = (h_1, h_2, \ldots, h_l)$, de comprimento l, escolhidos aleatoriamente na população, existe uma probabilidade ρ_r que os indivíduos gerem dois indivíduos novos. Para criar novos indivíduos a partir da carga genética de outros dois indivíduos, é aplicada uma operação adaptada do cruzamento de um ponto (*crossover* simples) (Mitchell, 1998).

Como cada cromossomo possui genes que representam diferentes níveis na árvore Υ, o operador de cruzamento é aplicado a cada grupo de genes de mesmo nível. Assim, para cada grupo de genes de mesmo nível, é escolhido um número $r \in \{1, 2, \ldots, l\}$ indicando o ponto de cruzamento. Por fim, duas novas cadeias são formadas a partir de C_1 e C_2 através da troca de um conjunto de atributos à direita da posição r, resultando em $C_1' = (g_1, \ldots, g_r, h_{r+1}, \ldots, h_l)$ e $C_2' = (h_1, \ldots, h_r, g_{r+1}, \ldots, g_l)$.

Exemplo 3.8 *O cruzamento dos cromossomos C_1 e C_2, escolhidos aleatoriamente na população, possui uma probabilidade ρ_r de gerar dois novos indivíduos. Considere, por exemplo, que C_1 e C_2 representam pesos em dois níveis de composição, como na figura 3.1. O cruzamento ocorre da seguinte forma:*

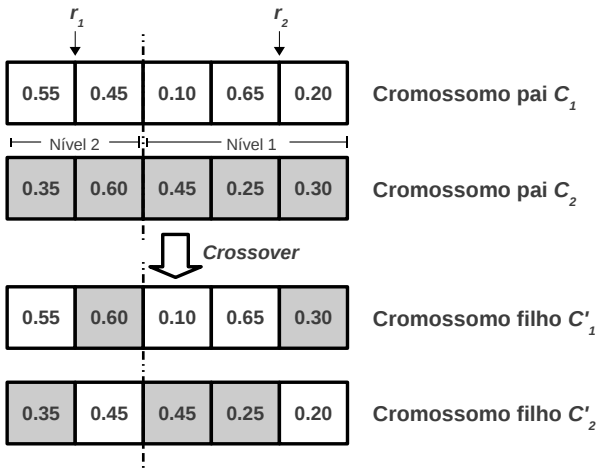

Figura 3.6: Exemplo de *crossover*

O operador cruzamento de um ponto mostra-se mais aplicado ao tipo de representação da solução empregado neste trabalho. Uma vez que a representação do indivíduo é feita em níveis, a utilização de um operador de dois pontos em cada nível poderia ocasionar a descaracterização dos descendentes a cada geração, dado que, para cada nível ter-se-ia um número muito maior de possibilidades de se combinar material genético que representa o mesma solução.

A cada novo cromossomo gerado pelo processo de cruzamento, existe uma probabilidade ρ_m de que o cromossomo sofra uma mutação. A mutação é um processo no qual um gene é aleatoriamente substituído (ou modificado) por outro, resultando em um novo cromossomo. O objetivo da mutação é impedir que a população se torne muito homogênea, fazendo com que o algorimo não consiga sair de um mínimo local. A mutação insere uma heterogeneidade que permite que o algoritmo explore outras áreas do espaço de soluções. Geralmente a probabilidade ρ_m é pequena.

Neste trabalho, é realizada uma mutação indutiva (Poli et al, 2008), na qual o cromossomo que será mutado tem um dos seus genes g escolhido aleatoriamente e, em seguida, é sorteado um valor $x \in \{-2, -1, 1, 2\}$ de tal forma que g recebe o novo valor $g' = g + x\tau$. Para evitar a geração de pesos negativos, introduzimos a restrição de que $g' \geq 0$.

Exemplo 3.9 *Seja um cromossomo* $C = (0.5, 0.1, 0.2)$, $\tau = 0.1$ *e* $p = 2$ *a posição do gene de* C *que sofrerá mutação. Supondo um número* $x = -2$ *sorteado aleatoriamente, o segundo gene de* C *será alterado para* $g' = 0.1 + (-2 \times 0.1) = -0.1$. *Como não é permitido genes com valores negativos, o*

processo se reinicia com a escolha de um novo gene g e de um novo valor x que resulte em $g + x\tau \geq 0$.

3.5 Intensificação da solução

Após a formação da população, todos os indivíduos são avaliados pela função de avaliação ϑ e o indivíduo com a melhor aptidão, ou seja, o indivíduo $C_v = min(\vartheta(\Xi, C))$ é denominado *indivíduo vencedor*. O indivíduo vencedor armazena a melhor solução existente na população.

Ao identificar o vencedor, realizamos um processo de intensificação da solução com a finalidade de encontrar uma melhor solução. Para tal, realizamos uma busca local com a solução vencedora.

Um algoritmo de busca local define, para cada solução, uma vizinhança composta por um conjunto de soluções com características muito próximas (Aarts & Lenstra, 2003). Dada uma solução corrente, uma das formas de implementar um algoritmo de busca local é percorrer a vizinhança dessa solução em busca de outra com valor menor (para um problema de minimização, como o apresentado neste trabalho). Se uma melhor solução for encontrada, torna-se a nova solução corrente e o algoritmo continua (Vieira, 2006).

Para selecionar uma melhor solução na vizinhança, basicamente três estratégias podem ser aplicadas (Talbi, 2009): (1) a estratégia da maior melhoria, (2) a estratégia da primeira melhoria ou (3) a estratégia da seleção randômica. A primeira estratégia consiste em percorrer a vizinhança de forma exaustiva e selecionar a melhor solução de toda a vizinhança. Este tipo de exploração pode ser custosa para vizinhanças muito grandes. A segunda estratégia consiste em selecionar a primeira solução gerada que seja melhor que a solução corrente. Assim, uma solução vizinha melhor que a corrente é imediatamente selecionada para substituir a solução atual. Por último, a estratégia randômica consiste em aplicar uma função de seleção randômica para escolher uma solução na vizinhança que seja melhor que a solução atual.

O algoritmo de busca local utilizado nesta abordagem consiste na busca de um $C_z = min(\vartheta(\Xi, C_z)$, onde C_z faz parte da vizinhança $V = V^+ \cup V^-$ e $C_z < C_v$. O conjunto V^+ é formado pelas soluções vizinhas de $C_v = (g_1, g_2, \ldots, g_n)$ que possuem características iguais às características de C com exceção de uma única característica igual a $g_i + \tau$. O conjunto de soluções vizinhas V^-, por sua vez, é formado por soluções que possuem características iguais às características de C com exceção de uma única característica de valor $g_i - \tau \geq 0$. Ou seja, $V^+ = \{(g_1 + \tau, g_2, \ldots, g_n), (g_1, g_2 +$

$\tau, \ldots, g_n), \ldots, (g_1, g_2, \ldots, g_n + \tau)\}$ e $V^- = \{(g_1 - \tau, g_2, \ldots, g_n), (g_1, g_2 - \tau, \ldots, g_n), \ldots, (g_1, g_2, \ldots, g_n - \tau)\}$.

Nesta abordagem, não é realizada uma busca exaustiva em toda a vizinhança da solução corrente, mas é selecionada a melhor solução presente em V. Como são verificados todas as soluções em V, a exploração de soluções em V é $O(n)$, onde $|V| \leq 2n$ e n é a quantidade de genes do cromossomo.

Caso seja encontrado um C_z, este é inserido na população e se torna o novo vencedor. Uma porcentagem t_v das melhores soluções presentes na vizinhança de C_v é inserida na população.

Este processo de intensificação é realizado a cada b gerações. A periodicidade b geralmente é determinada como 10% do número total de gerações. Ou seja, se o algoritmo for rodar por 300 gerações, então a periodicidade da busca local poderia ser determinado como uma vez a cada 30 gerações.

3.6 Construção da nova geração

O GNoSIS+ é um algoritmo genético elitista, o qual, a cada geração, preserva uma taxa de t_m melhores indivíduos. O esquema da figura 3.7 apresenta como novas gerações são geradas.

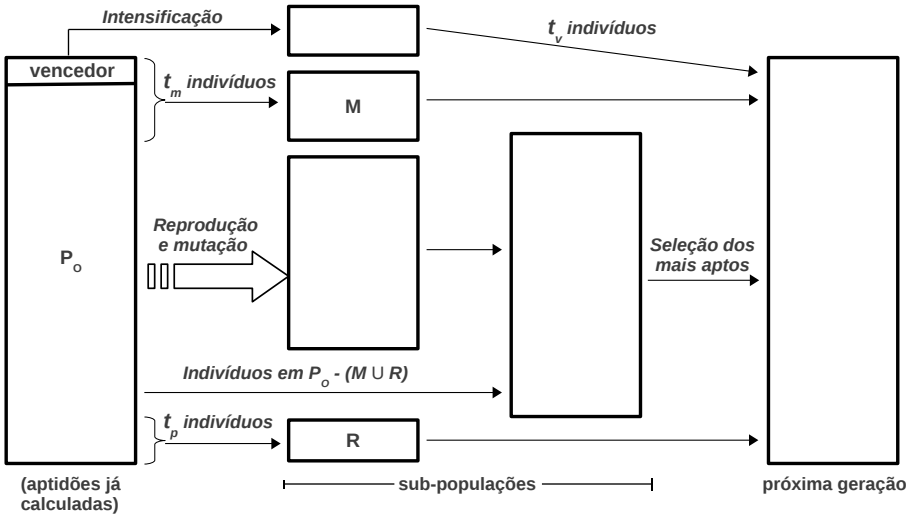

Figura 3.7: Como uma nova população é gerada

A partir de uma população P_0, a próxima geração é composa pelos t_m melhores indivíduos de P_0, formando o conjunto M. Para aumentar a diversificação genética, o conjunto R dos t_p piores indivíduos de P_0 também compõem a próxima geração. Os melhores indivíduos entre o conjunto formado pelos indivíduos gerados por reprodução na geração atual e os indivíduos intermediários, ou seja, as soluções que não estão dentre as melhores e

nem dentre as piores soluções da população P_0 (isto é, $P_0 - (M \cup R)$) são preservados para a nova geração. Por fim, quando é realizado o processo de intensificação, a nova população é composta, também, dos t_v melhores indivíduos que representam soluções vizinhas ao vencedor de P_0.

Para evitar que soluções persistam durante muitas gerações, como superindivíduos (Banzhaf et al, 2000) ou soluções muito ruins, é aplicado o conceito de mortalidade na população. Os indivíduos que atingem uma idade m determinada são descartados da nova geração.

O algoritmo 3 resume a abordagem implementada.

Algoritmo 3: Algoritmo genético implementado no GNoSIS+

Entrada: Parâmetros do sistema
Saída: $Vencedor$ (Melhor solução)
início
  Cria população inicial P;
  $Vencedor \leftarrow$ Melhor solução de P;
  enquanto *não é última geração* $\wedge\ \vartheta(\Xi, Vencedor) \neq 0$ **faça**
    Seleciona pares para cruzamento;
    para cada *par* (c, c') **faça**
      Cruzamento de (c, c'), com probabilidade ρ_r e probabilidade ρ_m de mutação;
    fim para cada
    se *Realizar Intensificação* **então**
      $Viz \leftarrow$ Vizinhança de $Vencedor$;
      se *melhor solução de* Viz *for melhor que* $Vencedor$ **então**
        $P \leftarrow P \cup \{$melhor solução de $Viz\}$;
      fim se
    fim se
    Retira indivíduos velhos de P;
    $P' \leftarrow (t_m\%$ melhores soluções de $P) \cup (t_p\%$ piores soluções de $P) \cup (t_v\%$ de $Viz) \cup$ (melhores soluções entre as soluções intermediárias de P e soluções geradas pelos cruzamentos);
    $P \leftarrow P'$;
    $Vencedor \leftarrow$ Melhor solução de P;
  fim enquanto
  Retorna $Vencedor$;
fim

3.7 Demonstração do comportamento da abordagem

Segundo Hoss & Stutzle (2007), algoritmos heurísticos são difíceis de analisar teoricamente, sobretudo algoritmos randômicos. Assim, segundo os autores, pesquisadores geralmente fazem uso de métodos empíricos para anal-

isar e avaliar algoritmos heurísticos. Embora algoritmos sejam especificados e matematicamente definidos, ainda assim, em muitos casos, este conhecimento é insuficiente para teoricamente derivar todos os aspectos relevantes do seu comportamento.

Algoritmos evolucionários, como a abordagem apresentada neste trabalho, possuem diversos parâmetros que controlam seu comportamento. O efeito dos vários parâmetros geralmente não são independentes, fazendo com que a busca pelos melhores valores de parâmetros se torne altamente custosa (Batiti, 1996). Contudo, mecanismos para adaptação dos valores dos parâmetros foram propostos na literatura (Coy et al, 2001; Birattari et al, 2002) e podem ser utilizados para substituir a análise empírica do comportamento do algoritmo.

No algoritmo proposto neste trabalho, porém, tais mecanismos não são aplicáveis para identificar os melhores valores de parâmetros para quaisquer instância do problema. Esta inaplicabilidade ocorre por razão da função objetivo utilizada neste trabalho. A função objetivo (equações 3-5 e 3-6) é dependente de diferentes artefatos de entrada do problema, isto é, dos exemplos de ocorrências fornecidas pelo engenheiro de ontologia, das funções de similaridade que serão utilizadas e das próprias ontologias que serão utilizadas. Com isso, cada novo artefato de entrada faz com que a função objetivo altere seus valores, isto é, cada nova instância do problema define uma nova função a ser aproximada. Assim, embora uma técnica de adaptação de valores de parâmetros possa ser aplicada, os valores encontrados somente serão válidos para a instância utilizada do problema.

Nesta seção é apresentada uma demonstração de como a abordagem heurística proposta se comporta em diferentes configurações. O ponto principal a verificar é como o número de gerações, o tamanho da população inicial e a periodicidade da busca local podem interferir nos resultados. Conforme pode-se inferir da equação 3-1, os valores das funções de similaridade, uma vez verificados, se tornam constantes de entrada para o algoritmo genético. Assim, ao invés de repassar correspondências reais para o algoritmo, são repassadas equações com constantes criadas aleatoriamente.

Todas as experimentações foram realizadas com os seguintes parâmetros definidos empiricamente:

- Taxa de seleção: 50%
- Probabilidade de cruzamento: 80%
- Probabilidade de mutação: 10%
- Reinserção dos melhores indivíduos: 30%

- Reinserção dos piores indivíduos: 10%
- Mortalidade: 5 gerações de vida
- Periodicidade da busca local: a cada 100 das gerações
- Inserção de vizinhança: 25% da vizinhança

Para se ter uma ideia mais precisa do comportamento dessa abordagem, é demonstrado na seção 3.7.1 o comportamento com um conjunto de equações simples de entrada e, na seção 3.7.2, o comportamento com um conjunto de equações simples e compostas. Por fim, a seção 3.7.3 demostra como o processo de intensificação da solução é benéfico para o resultado final.

Como a abordagem de algoritmo genético é uma abordagem probabilística, a solução final depende não só dos parâmetros de entrada, mas também da qualidade da população inicial que, neste trabalho, é gerada aleatoriamente. Assim, os gráficos apresentados nas seções seguintes representam a média dos valores encontrados em 10 execuções do algoritmo para cada configuração.

3.7.1 Demonstração de convergência com uma equação simples

Os resultados a seguir mostram os vencedores (melhor resultado) gerados pelo algoritmo genético a cada geração utilizando como entrada a equação 3-10. Os resultados foram coletados para 1000 gerações.

Observa-se na figura 3.8 o gráfico com os resultados gerados ao se rodar o algoritmo com populações iniciais de 10, 50, 100, 500 e 1000 indivíduos. O eixo Y do gráfico apresenta a aptidão ϑ do indivíduo vencedor.

$$\begin{cases} 0.5x + 0.1y + 0.2z = 1 \\ 0.6x + 0.8y + 0.1z = 1 \\ 0.1x + 0.5y + 0.8z = 1 \end{cases} \tag{3-10}$$

A figura 3.9 apresenta o gráfico gerado com os resultados a partir da geração 500, o qual permite uma melhor visualização da diferença entre os resultados para cada configuração da população inicial.

Pode ser verificado nas figuras 3.8 e 3.9 que há uma convergência muito rápida nas primeiras gerações e uma convergência mais lenta em gerações mais avançadas. Ainda, pode ser verificado que, confome maior for o tamanho da população inicial, melhor pode ser o resultado final gerado. Contudo, verifica-se que, para populações com 500 ou 1000 indivíduos, os valores finais médios são iguais, embora populações com 1000 indivíduos podem ter uma convergência inicial mais rápida do que populações com 500 indivíduos.

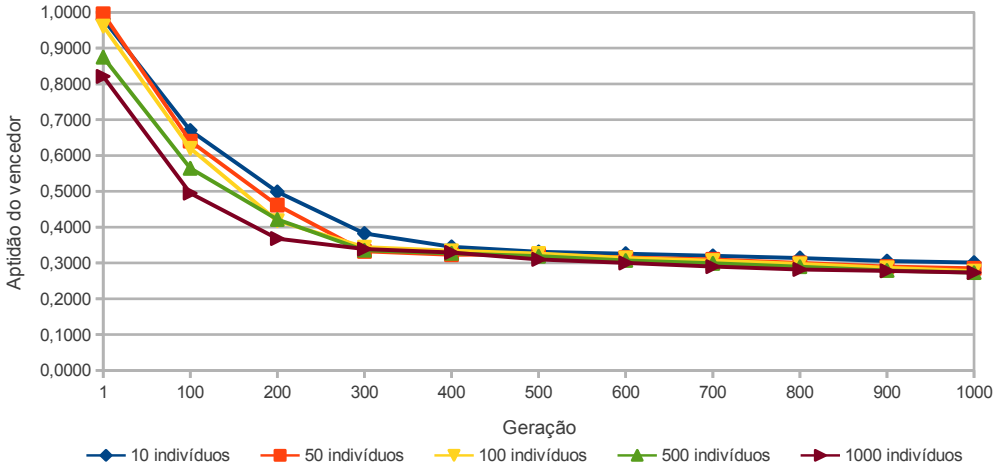

Figura 3.8: Demonstração da convergência do AG para a equação 3-10

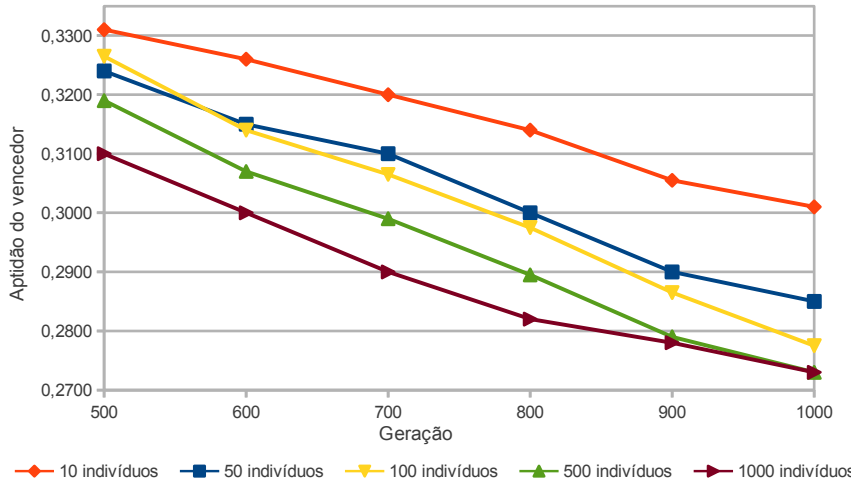

Figura 3.9: Gráfico ampliado da convergência do AG para a equação 3-10

3.7.2 Demonstração de convergência com uma equação composta

Os resultados a seguir mostram os vencedores (melhor resultado) gerados pelo algoritmo genético a cada geração utilizando como entrada a equação 3-11. Os resultados foram coletados para 1000 gerações.

Observa-se na figura 3.10 o gráfico com os resultados gerados ao se rodar o algoritmo com populações iniciais de 10, 50, 100, 500 e 1000 indivíduos. O eixo Y do gráfico apresenta a aptidão ϑ do indivíduo vencedor.

$$\begin{cases} f_{a_1} = 0,4x + 0,1y + 0,2z \\ f_{c_1} = f_{a_1}a + 0,1b \\ f_{c_1}c + 0,2d + 0,4e = 1 \\ \\ f_{a_2} = 0,3x + 0,2y + 0,4z \\ f_{c_2} = f_{a_2}a + 0,7b \\ f_{c_2}c + 0,1d + 0,1e = 1 \\ \\ f_{a_3} = 0,5x + 0,1y + 0,1z \\ f_{c_3} = f_{a_3}a + 0,4b \\ f_{c_3}c + 0,7d + 0,2e = 1 \end{cases} \tag{3-11}$$

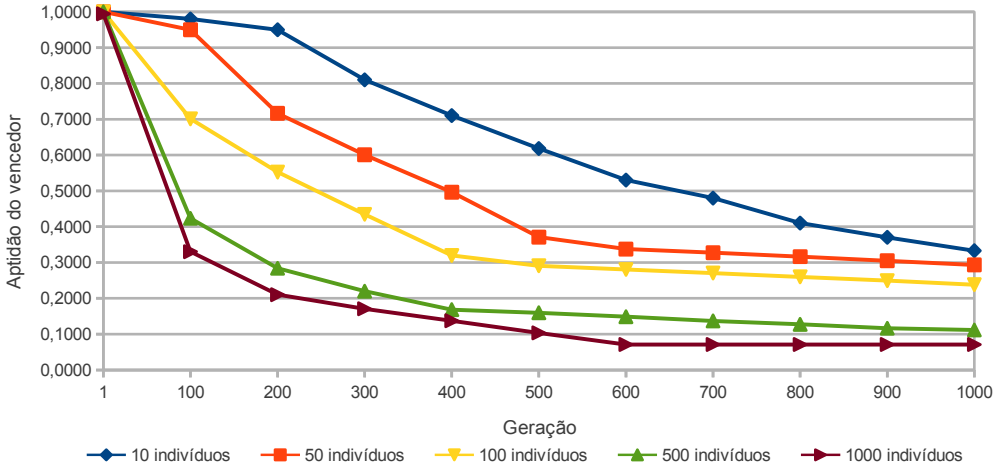

Figura 3.10: Demonstração da convergência do AG para a equação 3-11

A figura 3.11 apresenta o gráfico com os resultados a partir da geração 500, o qual permite uma melhor visualização da diferença entre os resultados para cada configuração da população inicial.

Pode ser verificado nas figuras 3.10 e 3.11 que não há uma convergência tão prematura quanto a verificada na figura 3.8, a não ser para populações com 500 e 1000 indivíduos. A principal diferença é a quantidade de pesos neste exemplo, o que faz com que o espaço de soluções seja bem maior do que do exemplo anterior. Além disso, como as equação são compostas, a alteração no valor de um gene faz com que ocorra um efeito em cascata nos valores das demais funções compostas. Ainda, também pode ser verificado que, confome maior for o tamanho da população inicial, melhor pode ser o resultado final gerado. Contudo, verifica-se que, para populações com 500 ou 1000 indivíduos, os valores finais médios são muito próximos, embora populações com 1000

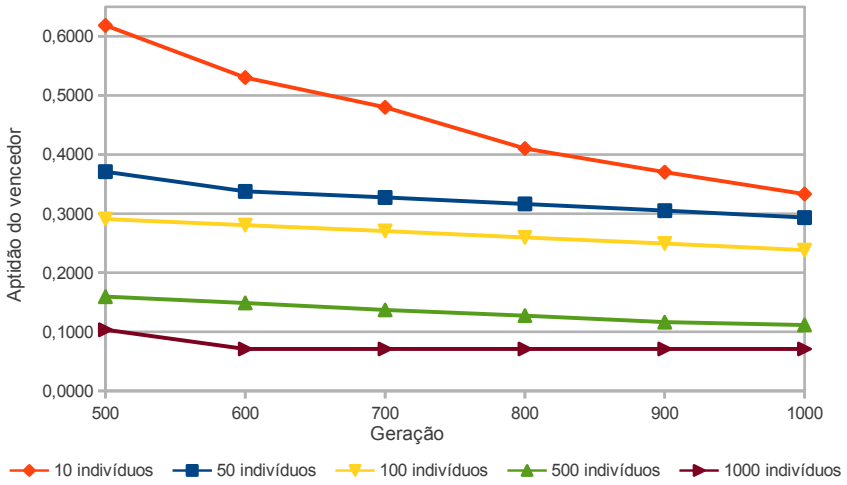

Figura 3.11: Gráfico ampliado da convergência do AG para a equação 3-11

indivíduos possam ter uma convergência inicial mais rápida do que populações com 500 indivíduos. Vale ressaltar, por fim, que a solução não se altera para população com 1000 indivíduos por volta da geração 600, o que pode trata-se de um ótimo local.

3.7.3 Demonstração do processo de intensificação da solução

Para verificarmos o quanto o processo de intensificação das soluções vencedoras pode melhorar o resultado alcançado, foram comparadas as aptidões médias dos vencedores nas dez execuções da seção 3.7.2 com criação máxima de 1000 gerações.

O objetivo dessa demonstração é verificar o quanto a aptidão da solução vencedora em uma certa geração pode melhorar após a realização da busca local. Para tal, foram registradas as aptidões dos vencedores antes do processo de intensificação e a aptidão da melhor solução da vizinhança, a qual se tornará o novo vencedor. A figura 3.12 apresenta o gráfico gerado com a porcentagem média de ganho na solução vencedora após a realização da busca local. Foram registrados os ganhos médios do vencedor de cada 100 gerações.

Como a abordagem de algoritmos genético é probabilística, os resultados alcançados a cada fase de intesificação podem se diferenciar a cada execução. Contudo, foram coletados resultados de 10 execuções do algoritmo de forma a encontrar uma variação média dos resultados.

Pode-se verificar na figura 3.12 que a busca local geralmente fornece soluções melhores do que a solução vencedora, visto que todas as gerações

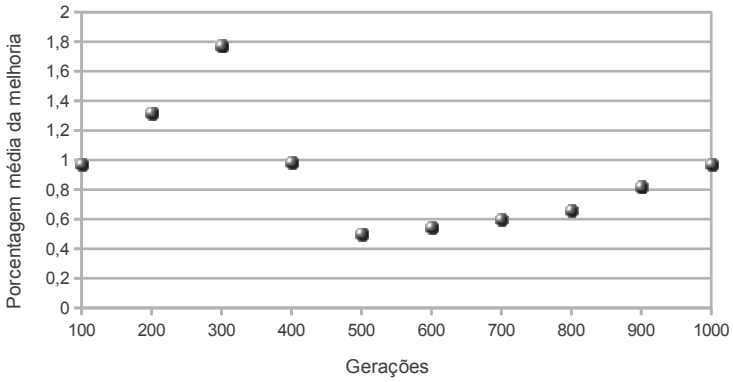

Figura 3.12: Demonstração da convergência do AG para a equação 3-11

possuem uma porcentagem média de ganho maior do que zero. Contudo, essa nova solução possui ganhos pequenos em relação à solução vencedora, visto que as porcentagens médias encontradas giram em torno de 0.5% a 1.5% de melhoria. Tal resultado se dá pelo fato de que a busca local é realizada como uma exploração da vizinhança da solução vencedora formada por pequenas variações em cada gene do cromossomo vencedor. Considerando o cromossomo vencedor como um vetor, a busca local pode ser considerada como um passo (de tamanho τ) em cada dimensão do vetor formado pelo cromossomo vencedor.

Vale ressaltar, contudo, que, embora não se registre uma porcentagem muito elevada de ganho na solução a cada processo de intensificação, o ganho acumulado é considerável, visto que o processo de intensificação permite uma convergência mais rápida da solução.