

Introdução aos Sistemas Imunológicos Artificiais

Grazziela Patrocínio Figueredo, Heder Soares Bernardino*
e Helio José Corrêa Barbosa

Resumo: Este capítulo descreve as principais abordagens empregadas nos sistemas imunológicos artificiais (SIAs). Como um novo campo da inteligência computacional, os SIAs vêm obtendo significativo sucesso desde o início da década de 90. Portanto, serão apresentados aqui alguns dos mais importantes algoritmos imuno-inspirados encontrados na literatura, tais como: (i) algoritmos de seleção negativa, (ii) redes imunológicas artificiais, (iii) algoritmos de seleção clonal e (iv) algoritmos baseados na teoria do perigo e células dendríticas. Embora sua inspiração natural sugira que os SIAs sejam aplicáveis a problemas de segurança computacional, abordaremos neste capítulo sua aplicação na pesquisa operacional (PO).

Palavras-chave: Sistemas imunológicos artificiais, Otimização, Pesquisa operacional.

Abstract: *This chapter describes the main approaches adopted in the artificial immune systems (AISs). This is a field in computational intelligence which has been achieving significant success since the early nineties. We present, therefore, some of the main immuno-inspired algorithms from the literature, based on (i) negative selection, (ii) artificial immune networks, (iii) clonal selection, and (iv) danger theory and dendritic cells. Although their natural inspiration suggests that AISs are applicable to computational security problems, here we will study their application in the operations research field.*

Keywords: *Artificial immune systems, Optimization, Operational research.*

1. Introdução

Já há algum tempo que a natureza vem servindo de inspiração para a solução de problemas computacionais. Em particular, os sistemas biológicos têm se destacado e, entre eles, está o sistema imunológico. Os Sistemas Imunológicos Artificiais (SIAs) interligam as áreas da imunologia, ciência da computação, matemática e engenharia, através de pesquisas envolvendo os aspectos teóricos, de modelagem e de aplicações.

O objetivo deste capítulo é introduzir as técnicas mais populares inspiradas no sistema imunológico e suas aplicações. Em especial, destaca-se a revisão da literatura desta classe de técnicas aplicadas a problemas de pesquisa operacional (PO), foco do presente livro.

2. Sistema Imunológico Natural

A vida é baseada em um frágil equilíbrio de milhões de reações químicas simultâneas e eventos inerentes a um ser vivo, como as interações deste com o ambiente e outros seres. Um acidente corriqueiro, como uma pequena farpa que penetre a pele de um indivíduo cria uma instabilidade no seu organismo. Neste caso, só existem duas possibilidades, ou o ser vivo reconhece que existiu uma agressão, o tipo de agressor e a melhor forma de combatê-lo, ou sucumbe. Ao aparato de reconhecimento, ação e controle do organismo frente às agressões bioquímicas dá-se o nome de Sistema Imunitário, Imunológico, ou também, Imune (Janeway et al., 2001).

O Sistema Imunológico (SI) é um dos sistemas biológicos mais complexos de que se tem conhecimento. Apesar da especificação da sua função ser a princípio bastante simples – detectar e eliminar qualquer organismo estranho – a execução desta tarefa não é trivial. Ainda hoje não se sabe exatamente como o SI funciona, nem mesmo qual a dimensão do seu papel dentro das interações do organismo (Golub, 1992). Entretanto, à luz do que atualmente é conhecido sobre a sua dinâmica, muito se tem feito na tentativa de construir sistemas artificiais inteligentes contendo algumas de suas propriedades (Hofmeyr & Forrest, 2000). Isto por duas razões: a primeira é tentar trazer para o computador partes do SI através de modelos mais rudimentares e simulações

*Autor para contato: hedersb@gmail.com

do seu funcionamento, no intuito de entender melhor as tarefas que ele executa. A segunda, utilizar o SI como inspiração para a construção de sistemas computacionais inteligentes.

Há muito interesse neste segundo tópico, pois existem propriedades buscadas por projetistas na construção de sistemas que estão naturalmente presentes no SI. Estas são robustez, tolerância a erros, reconhecimento de padrões, distribuição, memória, adaptabilidade, auto-organização, e autonomia.

Esta seção tem como objetivo discutir os conceitos fundamentais, as principais propriedades do SI e os mecanismos mais importantes das respostas imunes, a fim de fornecer subsídios para que o leitor entenda melhor os modelos artificiais apresentados neste capítulo.

2.1 Aspectos gerais das respostas imunológicas

Ao se tentar enxertar um pequeno fragmento de tecido vivo extraído de um animal em outro de espécie diferente, observa-se que em poucos dias este tecido será destruído e eliminado pelo organismo do indivíduo receptor. Este processo, conhecido como resposta imunológica, é caracterizado pela capacidade que o organismo possui de reconhecer e destruir material estranho, sem implicar em uma consequência fisiológica ou patológica de tal reação. Ao estudo das respostas imunológicas, suas consequências e mecanismos dá-se o nome de Imunologia (Janeway et al., 2001).

O exemplo de rejeição de tecido representa apenas uma das formas nas quais a resposta imunológica está presente. Entretanto, ele também é um indicador importante da existência de um sistema eficiente de detecção e eliminação de células e moléculas que se apresentam estruturalmente diferentes daquelas tidas como normais e pertencentes ao organismo. Portanto, existe uma espécie de “sistema de vigilância constante”, responsável pela identificação e remoção de células anormais, que podem ser pertencentes a outros seres vivos, como bactérias, ou mesmo células do próprio organismo contendo má formação ou defeitos, como por exemplo, aquelas presentes em tumores. Quanto mais eficaz for este sistema de localização e posterior remoção de irregularidades dentro do organismo, maiores são as suas chances de sobrevivência às invasões de parasitas e micro-organismos nocivos.

2.1.1 Imunidade inata e adaptativa

O SI dispõe de meios para se adaptar e tentar se proteger de agentes danosos – denominados antígenos – que se infiltram no organismo ao longo da existência do indivíduo. Entretanto, existem também mecanismos de defesa não condicionados a estímulos externos e que já fazem parte do organismo do indivíduo desde o seu nascimento. Esta é a chamada **imunidade natural** (nativa ou inata) e sua estrutura básica é inerente a toda uma espécie, e não a um exemplar apenas; é de caráter geral e proporciona um tipo de defesa amplo, porém, não discriminado. Fazem parte desta classe as barreiras físico-químicas da pele, membranas, saliva, lágrima, mucosas, enzimas digestivas, algumas células brancas e várias moléculas originadas no sangue (Abbas et al., 1998).

O outro tipo de imunidade é aquele que confere ao sistema imunológico a robustez de se proteger especificamente contra substâncias estranhas, a chamada **imunidade específica**, adaptativa ou adquirida. Este mecanismo de defesa é induzido ou estimulado pela exposição a fatores estranhos ao organismo.

Tanto na imunidade inata como específica, há duas classes distintas que se diferenciam pela forma como a resposta imune se dá. São elas a **resposta imunitária humoral** – presente no soro sanguíneo – e a resposta **mediada por células**.

Os elementos do soro sanguíneo que conferem resistência ao organismo a agentes ofensivos, como venenos e toxinas, são os chamados **anticorpos**, e sua produção é incitada pela substância danosa, ou **antígeno**. Um antígeno é qualquer molécula que pode ligar-se especificamente a um anticorpo. Seu nome deriva de sua propriedade de GERar ANTicorpos. Os anticorpos atuam reagindo quimicamente com os antígenos, de forma a neutralizar o seu princípio ativo e impedir que ele cause mais malefícios ao organismo. Os anticorpos são antígeno-específicos. Para um determinado antígeno, há um anticorpo produzido especificamente para se ligar a ele. Da mesma forma, quando o antígeno penetra o organismo, somente o anticorpo específico terá sua produção estimulada (Tizard, 1985).

Em uma primeira infestação de um certo antígeno no organismo (por exemplo, uma certa toxina), observa-se um período de latência (janela imunológica ou retardo) de vários dias, onde não há nenhum tipo de alteração no soro. Em seguida, surge uma alta concentração de anticorpos¹ que pode, no decorrer de alguns dias, atingir o seu grau máximo e cair rapidamente, o que caracteriza a chamada resposta imunitária primária. Se uma nova dose do antígeno for injetada após algum tempo decorrido desde a primeira reação, observa-se que o período de retardo é bem mais curto, e a quantidade de anticorpos aumenta rapidamente, caindo de forma lenta. Esta diferença entre as respostas primária e secundária indica a existência de um aparato produtor de anticorpos capaz de gravar, em uma forma de memória, exposições anteriores a antígenos, provendo o SI

¹ Quantidade determinada pela capacidade de neutralização de uma certa medida de antígenos.

da capacidade de se proteger mais rapidamente quando re-exposto a um agente danoso. Esta memória, em certos casos, pode ter sua dimensão reduzida com o passar do tempo. Entretanto, ela continua conferindo um tempo de latência mais curto em reações imunes secundárias, quando contrastado com as primárias.

A capacidade do organismo de reagir contra um tecido estranho, por exemplo, é dada através de células imunes chamadas células brancas. As células brancas do sistema inato mais importantes são as células fagocitárias, dentre as quais se destacam os macrófagos e as células dendríticas.

O papel das células dendríticas é capturar e apresentar o antígeno aos linfócitos, a fim de que ele seja reconhecido e combatido. Existem em sua superfície receptores capazes de reconhecer estruturas comuns a diversos tipos de antígenos. Assim, quando estes receptores encontram um antígeno, a célula dendrítica é estimulada a englobá-lo e degradá-lo. Ao ingerir o antígeno, a célula se ativa, amadurecendo em uma célula apresentadora de antígenos (APC) e migra até o linfonodo mais próximo. Já amadurecida, a célula é capaz de ativar os linfócitos (células brancas adaptativas) antígeno-específicos que determinam quando e como o sistema imune deverá responder aos agentes infecciosos. Pode-se dizer que os linfonodos são órgãos que atuam como “centros de inteligência” do SI, onde é procurado qual o linfócito mais apto a combater o antígeno apresentado pela APC (como as células dendríticas) e onde se dá o início à resposta ao agressor.

Linfócitos são células imunocompetentes recirculantes responsáveis pela imunidade específica. Estas células realizam um intercuro contínuo entre corrente sanguínea e linfa, e em seguida no sentido inverso, num processo chamado recirculação (Cormack, 1991). Durante o movimento de recirculação do sangue para a linfa e novamente para a corrente sanguínea, é provável que os linfócitos encontrem antígenos que possam ter penetrado o organismo. Um encontro de tal natureza pode, muitas vezes, desencadear uma resposta imune, pois este tipo de célula, em sua maioria, dispõe da capacidade funcional para reconhecer e responder a antígenos. Quando um pequeno linfócito reconhece um antígeno, ele aumenta de tamanho, replica o seu DNA e passa por uma série de divisões dando origem aos chamados clones, que são conjuntos de células identicamente programadas para o combate a um determinado agente agressor (Cormack, 1991). É possível destacar duas classes principais de linfócitos. Aqueles cuja diferenciação se dá no Timo são chamados linfócitos T e os outros cujo desenvolvimento se dá na medula óssea (*bone marrow*) – são os linfócitos B.

Os linfócitos T representam a maioria dos linfócitos do sangue e são os principais responsáveis pela imunidade adaptativa. Assim como os linfócitos B, eles tem sua origem na medula óssea. Entretanto, sua fase de maturação se dá no Timo. É nesta fase de maturação que o linfócito terá a sua função e especificidade antigênica definidos. Apesar do seu aspecto microscópico de maneira geral ser uniforme, os linfócitos T se subdividem em diversas classes funcionais distintas (Cormack, 1991).

Cabe aos linfócitos B a produção de anticorpos, sendo que cada clone de linfócitos B é responsável por um antígeno específico. Antes de sua ativação, cada linfócito B apresenta em sua superfície pequenas áreas de anticorpos específicas, também chamadas de parátomos. Estas áreas são responsáveis pelo reconhecimento do antígeno através da identificação do seu determinante antigênico, ou epítipo, que são as áreas contra as quais a resposta imunitária tende a se dirigir e onde os anticorpos costumam se ligar (Tizard, 1985). Assim, quando um linfócito B encontra e reconhece um antígeno, ele se torna ativado. Durante o seu processo de ativação, o linfócito aumenta, reproduz e sua progênie se diferencia em células secretoras de anticorpos, os plasmócitos, e algumas células de memória de mesma especificidade antigênica determinada pelos anticorpos da superfície celular do primeiro linfócito.

A explicação para que a capacidade de reconhecimento dos linfócitos B seja tão abrangente, cobrindo praticamente todos os antígenos existentes, está na forma como é estruturada uma molécula de anticorpo. O anticorpo é o receptor da célula B em sua forma solúvel, com sua molécula em formato de Y. O local de reconhecimento dos antígenos (parátomo), específico para cada tipo diferente de antígeno, fica localizado na ponta de cada um dos braços do Y. O repertório de anticorpos é gerado durante o desenvolvimento das células B por rearranjos do DNA (Janeway et al., 2001). Outro fator que potencializa esta diversidade são as hipermutações somáticas, que ocorrem nos linfócitos B ativados. As hipermutações, além de serem um mecanismo gerador de diversidade na população de anticorpos, são responsáveis também pelo refinamento da afinidade do anticorpo com relação ao seu antígeno específico. Elas operam sobre células B maduras localizadas nos órgãos linfóides secundários. Sua função é introduzir mutações pontuais nos parátomos, em uma taxa muito elevada (Janeway et al., 2001). O objetivo deste mecanismo é produzir anticorpos mutantes na superfície das células B, a fim de que algumas destas variações resultem em anticorpos de maior conformidade físico-química contra o antígeno quando comparados à anticorpo original. Aqueles linfócitos capazes de produzir anticorpos com poder de neutralização superior tendem a ser privilegiados no processo de escolha das células B que irão se diferenciar em plasmócitos. Este fenômeno de seleção dos melhores anticorpos é chamado maturação de afinidade.

2.2 Distinção entre próprio e não-próprio

O repertório de linfócitos presente no organismo é amplo o suficiente para reconhecer todos os possíveis invasores que apareçam ao longo de sua existência, o que é próprio do organismo (*self*), e o que é próprio e apresenta anomalias.

A especificidade dos linfócitos é determinada nas primeiras etapas de sua diferenciação, quando o RNA mensageiro codificador dos receptores das células B e T é montado a partir de segmentos gênicos rearranjados.

Dado o potencial criador dos rearranjos gênicos, associado aos outros fatores que geram diversidade, é possível admitir três classes de repertórios celulares (de Castro, 2001). O *repertório potencial* representa tudo que pode ser gerado pelos mecanismos de expressão genética associados às variações obtidas pelas mutações. O *repertório expresso* engloba moléculas disponíveis para tornarem-se receptores e o *repertório ativo* são os linfócitos que participam das respostas imunes.

Em uma fase seguinte à expressão do receptor antigênico na superfície do linfócito, há uma avaliação que verifica as propriedades de reconhecimento antigênico da célula com relação às moléculas do ambiente. Os resultados deste teste de especificidade e afinidade do receptor determinam se o linfócito está apto a sobreviver ou se deve ser eliminado.

2.3 Seleção positiva e seleção negativa

Em geral, linfócitos em desenvolvimento que interagem fracamente com antígenos próprios, ou auto-antígenos, recebem sinais de sobrevivência. A escolha dos linfócitos que permanecerão vivos é chamada seleção positiva. Particularmente para as células T, a seleção positiva escolhe aquelas cujos receptores são capazes de reconhecer e se ligar a moléculas de MHC-próprio (de Castro, 2001; de Castro & Timmis, 2002b), o que constitui uma premissa para que uma célula T seja capaz de armar uma resposta contra um antígeno. MHC é um complexo de genes que codifica várias proteínas receptoras trans-membranas celulares, que atuam no reconhecimento e na apresentação de constituintes dos agentes estranhos. Através do contato com estas proteínas, as células T iniciam a sua resposta imunológica ou estabelecem a tolerância ao antígeno.

A seleção positiva nas células B maduras ocorre após a hipermutação, quando a célula filha mutante é mais eficiente na neutralização do anticorpo e é preferida para expansão. Produtos ineficientes da hipermutação são eliminados do organismo.

Linfócitos de alta reatividade com antígenos próprios são sinalizados para morte, o que caracteriza a seleção negativa. Desta forma, linfócitos auto-agressivos são removidos do repertório antes de amadurecerem e desencadearem reações auto-imunes. Da mesma forma, antígenos que não apresentam malefícios ao organismo (como bactérias da flora intestinal, pelo de gato, etc.) em geral não são atacados pelo SI. Esta seleção estabelece a tolerância imunológica para auto-antígenos.

A sobrevivência dos linfócitos para a fase de ativação é condicionada a sinais enviados para os seus receptores. Aqueles que não receberem estes sinais são programados para a morte. A maior parte das combinações dos genes codificadores gera linfócitos auto-reativos. Desta forma, somente uma pequena parcela de linfócitos estão destinados a amadurecer e compor o repertório linfocitário do sistema adaptativo. Ainda assim, esta pequena fração restante é capaz de responder, em termos práticos, a qualquer antígeno apresentado (Janeway et al., 2001).

2.4 A teoria da seleção clonal

A teoria da Seleção Clonal (Burnet, 1957; Ada & Nossal, 1987) conjecturou que o antígeno, ao penetrar no corpo do hospedeiro, recrutaria um subconjunto de linfócitos capazes de reconhecê-lo. Linfócitos com maior reatividade com o antígeno seriam selecionados para expansão em clones em detrimento daqueles com menor capacidade de neutralização. Cada clone seria portador de um receptor único, ou seja, que não estaria presente em nenhum outro.

A seleção clonal inicia no momento em que um antígeno penetra o organismo e ativa o complexo inato. Aqueles linfócitos que apresentam maior poder de ligação com os antígenos se proliferam, na chamada expansão clonal. Ambas células B e T passam pelo processo de seleção clonal, com a diferença que linfócitos do tipo B sofrem hipermutações somáticas durante a expansão clonal. Aquelas células que apresentarem melhor ligação ao antígeno, sem serem auto-reativas, recebem estímulos para sobrevivência e proliferação, em detrimento de linfócitos menos aptos, que tendem a ser eliminados.

3. Técnicas Imuno-Inspiradas

O comportamento do sistema imunológico natural vêm inspirando o desenvolvimento de vários algoritmos computacionais. A Tabela 1 apresenta as classes de métodos imuno-inspirados mais comuns na literatura, bem como os tipos de problemas aos quais são normalmente aplicados.

Tabela 1. Classificação dos modelos computacionais imuno-inspirados comumente encontradas em aplicações de PO.

Algoritmo imuno-inspirado	Problema
Seleção Negativa	Deteção de anomalias, falhas e alterações
Rede imunológica	Aprendizagem de máquina e otimização
Algoritmos de seleção clonal	Busca e otimização
Teoria do perigo	Segurança de redes
Células Dendríticas	Segurança de redes, aprendizagem de máquina e robótica

As sub-seções que se seguem descrevem os algoritmos não em ordem cronológica, mas sim visando (i) apresentar primeiro e mais detalhadamente as técnicas imuno-inspiradas mais comumente encontradas na literatura de PO (algoritmos de seleção clonal e rede imunológica), e (ii) facilitar o entendimento do leitor, já que algumas técnicas mais recentes incrementam os métodos de seleção clonal com ideias da rede imunológica.

3.1 Algoritmos de seleção clonal

As técnicas que simulam o princípio de seleção clonal são aplicáveis a problemas de aprendizagem de máquina, reconhecimento de padrões e otimização (Bernardino & Barbosa, 2009). Como a maioria dos casos de PO recaem em situações onde se deseja obter uma solução ótima, o presente capítulo se concentra na aplicação dos algoritmos de seleção clonal a problemas de otimização. Além disto, a otimização é a principal área em que os algoritmos de seleção clonal são aplicados atualmente. Outras aplicações dos algoritmos de seleção clonal podem ser encontradas no trabalho de de Castro & Timmis (2002c).

Ao invés de reconhecer uma população de antígenos (como ocorreria em problemas de reconhecimento de padrões), quando se busca resolver um problema de otimização o objetivo é substituído por otimizar uma função (ou várias delas, no caso multi-objetivo). A ideia inspiradora desses métodos é a que somente células com capacidade de reconhecer antígenos irão sobreviver e gerar descendentes, ou seja, o processo geralmente ocorre como a maturação do sistema imunológico. A principal diferença é que, ao contrário do que ocorre na natureza, todos os passos de evolução dos linfócitos são aplicados apenas em anticorpos, já que normalmente as técnicas computacionais não fazem distinção entre anticorpos e linfócitos, tornando o sistema mais simples de ser representado.

3.1.1 CLONALG

Imitando o princípio de seleção clonal, o CLONALG (*CLOnal selection ALGORITHM*, ou algoritmo de seleção clonal) foi proposto por de Castro & von Zuben (2000, 2002) como um método a ser aplicado tanto em otimização, como será detalhado aqui, quanto a problemas de aprendizagem de máquina. O processo de melhora das soluções candidatas no CLONALG segue as etapas de clonagem destas soluções, hipermutação das novas células geradas e seleção daquelas com maior afinidade em relação ao antígeno (melhores soluções em relação ao objetivo).

A primeira versão do CLONALG, denominada CSA, foi proposta por de Castro & von Zuben (2000) e aplicada com sucesso a problemas de otimização multimodal e a uma instância de 30 cidades do problema do Caixeiro Viajante. Todavia, o CLONALG é hoje largamente referenciado pelo estudo apresentado em de Castro & von Zuben (2002), em que a sensibilidade em relação aos seus parâmetros e seu desempenho em um conjunto de problemas-teste foram analisados.

Um pseudo-código do CLONALG, elaborado com base no fluxograma e descrição contidos em de Castro & von Zuben (2002), pode ser encontrado no Algoritmo 1. Os parâmetros de entrada do usuário são: um valor β que determina o número de clones gerados por cada anticorpo (Equação 1), um fator ρ que ajusta a intensidade da hipermutação aplicada aos clones (Equação 2), o tamanho *tamanhoPopulacao* de soluções candidatas da população *anticorpos*, o número *nAleatorio* de anticorpos de menor afinidade que devem ser substituídos por novos anticorpos aleatoriamente gerados e a quantidade *nSelecao* de anticorpos a serem selecionados para a clonagem. Note que *nAleatorio* pode ser definido como uma proporção do tamanho da população, mesmo tendo sido inicialmente proposto como um parâmetro absoluto. Apesar de geralmente utilizado com código de Gray, como originalmente proposto, estudos também podem ser encontrados adotando codificação real (Garrett, 2004; Bernardino et al., 2011). As variáveis *afinidades*, *afinidadesNormalizadas* e *afinidadesClones* armazenam a qualidade das soluções candidatas (*anticorpos*), esses valores normalizados e a qualidade dos clones hipermutados, respectivamente.

Além das variáveis de entrada definidas previamente, o pseudo-código também utiliza as seguintes funções/métodos: “inicializaPopulacao”, cria (aleatoriamente) uma nova população de soluções candidatas;

Algoritmo 1: Um pseudo-código do CLONALG quando aplicado a problemas de otimização.

Entrada: $\beta, \rho, tamanhoPopulacao, nAleatorio, nSelecao$

Saída: *anticorpos*

```

1 início
2   anticorpos  $\leftarrow$  inicializaPopulacao(tamanhoPopulacao);
3   afinidades  $\leftarrow$  avalia(anticorpos);
4   enquanto critério de parada não é atingido faça
5     afinidadesNormalizadas  $\leftarrow$  normaliza(afinidades);
6     selecionados  $\leftarrow$  seleciona(anticorpos, afinidadesNormalizadas, nSelecao);
7     clones  $\leftarrow$  clona(selecionados, afinidadesNormalizadas,  $\beta$ );
8     hipermuta(clones, afinidadesNormalizadas,  $\rho$ );
9     afinidadesClones  $\leftarrow$  avalia(clones);
10    substituiPopulacao(anticorpos, afinidades, clones, afinidadesClones, nSelecao, nAleatorio);

```

“avalia”, avalia as soluções em relação à função objetivo; “normaliza”, normaliza as afinidades em $[0; 1]$; “seleciona”, seleciona os anticorpos que serão clonados; “clona”, clona as soluções candidatas; “hipermuta”, aplica o operador de hipermutação nos clones gerados; “substituiPopulacao”, substitui os piores *nSelecao* anticorpos pelos melhores clones gerados e gera *nAleatorio* novas soluções candidatas aleatoriamente também substituindo-as na população (cálculos adicionais da função objetivo são requeridos aqui).

Não é difícil perceber que, além do Algoritmo 1 e das descrições apresentadas anteriormente, especificações de como calcular o número de clones gerados por cada anticorpo e de como hipermutar estes clones ainda são necessárias para que se possa implementar uma versão do CLONALG para otimização. Segundo de Castro & von Zuben (2002), o número de clones n_{clones} que o anticorpo *anticorpo_i* gera é definido pela Equação (1):

$$n_{clones}(anticorpo_i) = arredonda \left(\frac{\beta \cdot |anticorpos|}{i} \right), \quad (1)$$

onde *arredonda* é um operador que retorna o valor inteiro mais próximo do seu argumento e *i* é a posição do anticorpo *anticorpo_i* na lista ordenada de forma decrescente pela afinidade normalizada, ou seja, o primeiro anticorpo da lista corresponde à melhor solução candidata. Também segundo a mesma referência, a hipermutação é aplicada com uma taxa α proporcional à afinidade do anticorpo que deu origem ao clone *clone_i*, como definida na Equação (2):

$$\alpha(clone_i) = \exp(-\rho \cdot \bar{f}_i), \quad (2)$$

com \bar{f}_i sendo a afinidade normalizada do anticorpo *i* correspondente.

A adaptabilidade a diversos tipos de problemas é a principal vantagem do CLONALG. Um algoritmo evolutivo, por exemplo, precisa ser estendido com alguma técnica de nicho (Bäck et al., 2000), tais como *Fitness sharing* (Goldberg & Richardson, 1987) e *Crowding* (Mahfoud, 1995), quando aplicado a problemas de otimização multimodais, enquanto que o CLONALG não requer tais técnicas adicionais para manter a diversidade de suas soluções. Apenas os seguintes ajustes precisam ser realizados (de Castro & von Zuben, 2002):

- $nSelecao = |anticorpos|$, ou seja, todos os anticorpos da população são selecionados para o processo de clonagem;
- n_{clones} é igual para todas as soluções candidatas;
- a população é atualizada substituindo-se os anticorpos pelo seu melhor clone quando uma melhora ocorre (este item foi inicialmente utilizado em um algoritmo imuno-inspirado no trabalho de Kelsey & Timmis (2003) e é hoje largamente utilizado tanto no CLONALG quanto em outros algoritmos de seleção clonal).

CLONALG é um dos algoritmos imuno-inspirados mais conhecidos para resolver problemas de otimização e suas ideias podem ser facilmente encontradas em aplicações de PO (de Castro & von Zuben, 2000, 2002; Agarwal et al., 2007; van Peteghem & Vanhoucke, 2009; van Peteghem & Vanhoucke, 2013), como poderá ser verificado na Seção 4.

3.1.2 opt-IMMALG

O algoritmo imuno-inspirado para problemas de otimização, opt-IMMALG, pode ser facilmente encontrado na literatura. Esta técnica foi originalmente proposta por [Cutello & Nicosia \(2002\)](#) como IA (*Immune Algorithm*), mas outros nomes foram elaborados enquanto melhorias e diversidade de aplicações do método imuno-inspirado iam sendo desenvolvidas. Neste progresso, pode-se citar o trabalho de [Cutello et al. \(2003\)](#) como o primeiro a apresentar o algoritmo com as características que possui atualmente (como a presença de um operador de idade). [Cutello et al. \(2005a\)](#) compararam o então chamado opt-IA com o CLONALG. [Cutello et al. \(2005b\)](#) elaboraram a versão com codificação real opt-IMMALG e [Cutello et al. \(2007\)](#) incluíram uma busca local específica do problema no então chamado IMMALG. No artigo de [Pavone et al. \(2012\)](#) foram definidas as nomenclaturas de opt-IMMALG e opt-IMMALG01, respectivamente, para as versões com codificação real e binária. Por fim, ressalta-se o MIA (*Memetic Immunological Algorithm*) uma versão deste algoritmo que foi analisada em um problema de alocação de recursos em uma rede de comunicação.

O MIA ([Costanza et al., 2011](#)) inclui uma busca local em seu processo de melhora das soluções candidatas. A primeira versão do algoritmo, encontrada em [Cutello & Nicosia \(2002\)](#), é similar ao CLONALG (veja Seção 3.1.1), possuindo inclusive um passo no procedimento responsável por gerar aleatoriamente novas soluções candidatas a cada iteração, denominado operador de nascimento. Um operador de idade, utilizado pelas versões atuais do método, bem como um operador de busca local e critério de parada específicos do problema foram adicionados ao método em [Cutello et al. \(2003\)](#). [Cutello et al. \(2004\)](#) analisaram diversos operadores de hipermutação para o algoritmo enquanto que [Cutello et al. \(2005b\)](#) elaboraram sua versão com codificação real. Uma versão atual do método, como apresentada em [Pavone et al. \(2012\)](#), é apresentada no Algoritmo 2.

Algoritmo 2: Um pseudo-código do opt-IMMALG.

Entrada: $dup, \rho, \tau_B, tamanhoPopulacao$

Saída: $anticorpos$

1 **início**

2 $anticorpos \leftarrow inicializaPopulacao(tamanhoPopulacao);$

3 $afinidades \leftarrow avalia(anticorpos);$

4 **enquanto** critério de parada não é atingido **faça**

5 $afinidadesNormalizadas \leftarrow normaliza(afinidades);$

6 $clones \leftarrow clona(anticorpos, dup);$

7 $hypermutate(clones, afinidadesNormalizadas, \rho);$

8 $afinidadesClones \leftarrow avalia(clones);$

9 $substituiPopulacao(anticorpos, afinidades, clones, afinidadesClones, tamanhoPopulacao, \tau_B);$

Além de ρ e $tamanhoPopulacao$, já observados no Algoritmo 1, o Algoritmo 2 utiliza parâmetros referentes ao número de clones gerados por cada anticorpo (dup) e uma idade máxima para as soluções candidatas (τ_B). As funções utilizadas no pseudo-código do algoritmo opt-IMMALG foram previamente descritas na Seção 3.1.1. Entretanto, existem algumas diferenças em suas definições que serão apresentadas a seguir.

A primeira diferença que pode ser observada entre os algoritmos CLONALG e opt-IMMALG está na função “clona”, que neste último gera por padrão dup clones para quaisquer anticorpos da população corrente. Além disto, como o opt-IMMALG inclui um operador de idade (função “substituiPopulacao”) os novos anticorpos possuem um atributo correspondente à sua idade que deve ser inicializado com um valor aleatório uniforme em $[0; \frac{2}{3}\tau_B]$ ([Pavone et al., 2012](#)). Outra pequena mudança ocorre na função “normaliza” da versão com codificação real do opt-IMMALG, onde a normalização é feita em $[0; 1]$ utilizando a melhor função objetivo encontrada até o momento decrescida de $\theta\%$, que é um parâmetro a ser definido pelo usuário.

A hipermutação aqui também pode ser aplicada de maneira diferente. Ao invés de utilizar uma taxa de mutação, como acontece no CLONALG, aqui α é utilizado para indicar o número de elementos M a serem modificados na solução candidata, como definido pela Equação (3):

$$M(\alpha) = \lfloor \alpha l + 1 \rfloor, \quad (3)$$

tomando l como a dimensão da representação do anticorpo (seja binária ou real, onde l é igual à dimensão do problema). Além da Equação 2, α também pode ser calculado utilizando a Equação (4):

$$\alpha(clone_i) = \frac{\exp(-\bar{f}_i)}{\rho}, \quad (4)$$

que usa parâmetros já definidos. É importante destacar que a utilização desta equação como meio de definir o potencial do operador de movimento de um algoritmo imuno-inspirado foi proposta por [de Castro & Timmis \(2002a\)](#) (veja Seção 3.2). Para o caso em que a codificação da solução candidata é real, os elementos i de um clone são modificados como definido pela Equação (5):

$$\alpha(\text{clone}_i) = (1 - \beta) \text{clone}_i + \beta \text{clone}_j, \quad (5)$$

onde $\beta \in [0; 1]$ é um número aleatoriamente gerado de uma distribuição uniforme, $1 \leq i \neq j \leq l = n$ e n é a dimensão do problema.

Tem-se por fim a função “*substituiPopulacao*”, que no opt-IMMALG inclui inicialmente um operador de idade, responsável por eliminar soluções candidatas antigas. Existem dois tipos de operadores de idade: (i) estático, que remove da população anticorpos com idade superior a τ_B ; e (ii) elitista, em que a melhor solução é mantida mesmo com idade superior a τ_B . A próxima população de soluções candidatas será composta pelas melhores soluções restantes, ou seja, pelos anticorpos de melhor afinidade não eliminados pela idade.

O MIA ([Costanza et al., 2011](#)), que é a versão aplicada ao problema de alocação de recursos, inclui a mesma busca local utilizada em [Cutello et al. \(2003\)](#), bem como seus custos adicionais de avaliação da função objetivo, entre as linhas 8 e 9 do Algoritmo 2.

3.1.3 Outros algoritmos de seleção clonal

Diversos outros algoritmos inspirados pela teoria da seleção clonal já foram propostos como métodos de otimização; em particular para problemas de PO. Em geral, a diferença entre estas técnicas reside nas definições de cada etapa da evolução dos linfócitos do sistema imunológico.

O BCA (*B-Cell algorithm*, ou Algoritmo da Célula B) foi proposto por [Kelsey & Timmis \(2003\)](#) e, mesmo sendo muito parecido com o CLONALG, utiliza um operador de hipermutação ímpar. Ao invés da modificação na solução candidata ocorrer em múltiplos alelos (*bits*) aleatoriamente selecionados, o chamado operador de mutação contígua aplica a mutação sobre um trecho (contínuo e de tamanho escolhido aleatoriamente) da representação binária da solução candidata. O BCA também define o operador de metadinâmica que opera selecionando um dos clones e modificando-o aleatoriamente. Nota-se que esse operador é similar à geração aleatória de novas soluções candidatas do CLONALG (simulando a medula óssea). Entretanto, o operador de metadinâmica é utilizado como um operador de movimento adicional, não adicionando cálculos da função objetivo. Finalmente, é importante destacar que por padrão o processo de substituição da população no BCA troca a solução candidata corrente por seu melhor clone hipermutado, no caso de haver melhora. Atualmente, esse modelo de substituição é normalmente adotado no CLONALG.

3.2 Redes imunológicas

Outros modelos, como Modelo B e *Shape Space*, também são baseados nesta ideia que serviu, além disto, para o desenvolvimento de diversos algoritmos computacionais para a resolução de problemas de análise de dados, classificação, agrupamento e otimização.

[de Castro & von Zuben \(2001\)](#) propuseram um algoritmo para análise de dados inspirado pela teoria da rede imunológica chamado aiNet. Uma versão aplicável a problemas de otimização foi posteriormente elaborada por [de Castro & Timmis \(2002a\)](#). O então chamado opt-aiNet é uma extensão do CLONALG (veja a Seção 3.1.1) com codificação tipicamente real e incluindo no procedimento de melhora das soluções candidatas um passo que envolve a interação entre as células da rede. No opt-aiNet, uma rede é formada pelas distâncias entre as soluções candidatas e um processo de supressão elimina as piores soluções candidatas entre as consideradas similares. Assim, a população não mais possui tamanho pré-fixado. Um pseudo-código do opt-aiNet é apresentado no Algoritmo 3.

É fácil verificar que os Algoritmos 1 e 3 são muito similares. À primeira vista, é possível notar que o algoritmo do opt-aiNet não possui o parâmetro $n_{Selecao}$ nem a função “*seleciona*” pois todas soluções candidatas são selecionadas (como na adaptação a problemas multi-modais apresentada na Seção 3.1.1). Além disto, α é calculado utilizando a Equação 4 e o operador de hipermutação é agora aplicado a soluções candidatas com codificação real, como definido pela Equação (6):

$$\text{clone}_i = \text{clone}_i + \alpha N(0, 1), \quad \forall 1 \leq i \leq n, \quad (6)$$

onde $N(0, 1)$ é uma variável aleatória Gaussiana de uma distribuição com média zero e desvio padrão unitário.

A função “*substituiPopulacao*” inicia pela inclusão de todos os clones hipermutados na população e segue com a criação da rede imunológica calculando a distância entre todas as células deste conjunto. As piores soluções candidatas entre as ditas similares (com base no parâmetro σ) são suprimidos da rede. É importante ressaltar que [de Castro & Timmis \(2002a\)](#) propuseram o uso do espaço de afinidade para verificação da

Algoritmo 3: Um pseudo-código do opt-aiNet.**Entrada:** $\beta, \rho, tamanhoInicialPopulacao, nAleatorio, \sigma$ **Saída:** *anticorpos*

```

1 início
2   anticorpos  $\leftarrow$  inicializaPopulacao(tamanhoInicialPopulacao);
3   afinidades  $\leftarrow$  avalia(anticorpos);
4   enquanto critério de parada não é atingido faça
5     afinidadesNormalizadas  $\leftarrow$  normaliza(afinidades);
6     clones  $\leftarrow$  clona(anticorpos, afinidades,  $\beta$ );
7     hipermuta(clones, afinidadesNormalizadas,  $\rho$ );
8     afinidadesClones  $\leftarrow$  avalia(clones);
9     substituiPopulacao(anticorpos, afinidades, clones, afinidadesClones, nAleatorio,  $\sigma$ );

```

similaridade, mas outros trabalhos como o de Lee & Zomaya (2008) utilizam distância no espaço de busca. Por fim, novas soluções candidatas são aleatoriamente geradas e incluídas na população.

Novas versões bem como diferenças entre implementações do opt-aiNet podem ser observadas na literatura. Um exemplo refere-se a quando utilizar o operador de supressão e de geração de novas soluções candidatas. Apesar de alguns trabalhos considerarem ambos passos em todas as iterações (Kelsey & Timmis, 2003; Lee & Zomaya, 2008), a versão proposta originalmente por de Castro & Timmis (2002a) apenas o faz quando não se observa mais melhora média das soluções candidatas.

3.3 Outras técnicas imuno-inspiradas

Além daquelas técnicas apresentados nas Seções 3.1 e 3.2, existem muitos outros métodos que são inspirados pela ideia da seleção negativa, teoria do perigo e das células dendríticas. Entretanto, dada a sua maior aplicabilidade a problemas de detecção e segurança, escapando assim ao escopo de pesquisa operacional, este capítulo apenas as descreve brevemente.

3.3.1 Seleção negativa

Os métodos computacionais baseados neste processo são denominados algoritmos de seleção negativa e estão entre os métodos imuno-inspirados mais antigos, aplicados a vários tipos de problemas. O algoritmo, aplicado ao reconhecimento de anomalias, inicia-se com uma etapa geracional, onde os detectores são gerados por um processo aleatório e depois testados contra amostras próprias. Caso um detector defina o próprio como anomalia, este é eliminado da população de detectores. Na segunda etapa, chamada de detecção, a coleção de detectores é utilizada para discriminar o que é próprio e o que é anômalo.

3.3.2 Teoria do perigo

Esta teoria introduzida por Polly Matzinger (Matzinger (1994)) inspirou pesquisadores a desenvolverem sistemas computacionais para a segurança de redes (Aickelin & Cayzer (2002)). O princípio é que deve haver uma célula apresentadora de antígenos, capaz de capturar e apresentar o sinal de perigo apropriado. Além disto, é necessário estabelecer o que é um sinal de perigo e como este perigo se modifica ao longo do tempo. É preciso também estabelecer as medidas de tempo e espaço para a implementação de um algoritmo baseado nesta inspiração. Os esforços em pesquisas nesta área culminaram com o desenvolvimento dos algoritmos de células dendríticas.

3.3.3 Algoritmos de células dendríticas

Os algoritmos de células dendríticas são baseados em um modelo abstrato do comportamento das células dendríticas humanas (Greensmith et al. (2005)). Constituem-se de componentes baseados nas principais propriedades de reconhecimento destas células, dentre as quais se destacam a fusão de dados e correlação. Foram inicialmente concebidos como parte do projeto da teoria do perigo para a detecção de anomalias na área de segurança de redes e depois suas potencialidades foram expandidas a diversas áreas, como aprendizado de máquina e robótica.

4. Aplicações em Pesquisa Operacional

4.1 Agendamento de tarefas

O problema de agendamento ou programação de tarefas consiste em alocar recursos limitados visando a realização de um conjunto de tarefas dentro de certo intervalo de tempo. Torna-se um problema de otimização quando se busca o melhor agendamento das atividades. Diferentes funções objetivo podem ser consideradas, tais como minimizar o tempo total gasto (*makespan*) ou minimizar penalidades por adiantamentos e atrasos em relação às datas de entrega das tarefas. O JSSP (*job-shop scheduling problem*) consiste em n tarefas a serem processadas por m máquinas. Cada tarefa tem operações que atendem a relações de precedência pré-definidas. Ao processamento da j -ésima tarefa na r -ésima máquina denota-se como a operação O_{jr} que usará esta máquina durante o período de processamento p_{jr} . Uma programação é então um conjunto de tempos de processamento c_{jr} para cada operação satisfazendo as restrições existentes, e um objetivo comum é a minimização do *makespan*.

A aplicação de SIAs a problemas deste tipo parece ter iniciado com Hart et al. (1998) e Hart & Ross (1999). Aqui abordaremos o trabalho de Coello et al. (2003), baseado em dois mecanismos dos SIAs: (i) o princípio da seleção clonal; (ii) cada anticorpo é formado pela concatenação de segmentos oriundos de uma biblioteca gênica. Cada elemento desta biblioteca representa um sequência de tarefas processadas por uma máquina e um anticorpo é uma cadeia com a sequência de tarefas processadas por cada máquina. A representação adotada é a de permutações com repetições proposta por Yamada & Nakano (1997). Além disto, o procedimento inclui um mecanismo de busca local que busca alocar tarefas aproveitando eventuais espaços disponíveis na programação de cada máquina.

O algoritmo inicia gerando aleatoriamente uma biblioteca de anticorpos e, em seguida, um antígeno (solução candidata) também aleatoriamente. Combinando segmentos tomados da biblioteca, um único anticorpo é gerado, decodificado e submetido a uma busca local que visa eliminar hiatos entre as tarefas de uma dada máquina. Se a solução codificada por este anticorpo for superior à codificada pelo antígeno este passa a armazenar o conteúdo do anticorpo. Em seguida N clones são gerados e submetidos à mutação. Os melhores segmentos destas soluções são então usados para atualizar a biblioteca. Neste ponto são feitas comparações entre as sequências de cada máquina com aquelas que compõem o antígeno, procurando pelas melhores soluções parciais para tentar recombiná-las, gerando um novo anticorpo. Observe que a representação adotada torna difícil a definição de uma medida de afinidade entre soluções e, assim, o valor da função objetivo (*makespan*) é usado para tal. Mais detalhes e um exemplo de aplicação podem ser encontrados em (Coello et al., 2003).

SIAs mais complexos para os problemas de sequenciamento de tarefas vêm sendo desenvolvidos ao longo dos anos como, por exemplo, em (Luh & Chueh, 2009) e (Bagheri et al., 2010), para o caso de *job shop* flexível. O caso de mais de um objetivo (*makespan* e *earliness/tardiness*, por exemplo) também vem sendo abordado via SIAs (Gao, 2010).

4.2 Problemas de roteamento

Várias problemas reais em PO podem ser enquadrados como problemas de roteamento e a literatura contém diversas aplicações de SIAs a estes casos.

Em (Keko et al., 2003) a ideia de vacinação é introduzida para melhorar o desempenho de um algoritmo genético no problema do caixeiro viajante. A vacinação corresponde ao uso de informação prévia sobre o problema. No caso, pares de cidades conectadas pelas arestas mais curtas (menos custosas) são determinadas e “inoculadas” nas soluções candidatas partindo da hipótese de que tais arestas tem maior probabilidade de estarem presentes na solução ótima. As soluções obtidas nesta primeira fase da busca podem ser melhoradas com o uso da heurística de Lin-Kernighan (Lin & Kernighan, 1973). Uma possível sofisticação do processo corresponde a gerar e modificar as vacinas durante a busca e selecionar as mais eficazes.

Ma et al. (2009) aplicam um algoritmo de seleção clonal para a resolução do problema de roteamento de um grupo de veículos que devem partir de um depósito central e atender às demandas de vários clientes de maneira a minimizar a distância total percorrida pelos veículos. Além disto, as restrições de capacidade e de comprimento de rota de cada veículo devem ser satisfeitas bem como a restrição de que a demanda de cada cliente deve ser atendida por um único veículo.

Um problema prático complexo é o planejamento de redes de distribuição de energia elétrica. A sua modelagem e resolução podem ser feitas a partir do problema de roteamento de veículos com múltiplos depósitos. Keko et al. (2004) mostram como este problema pode ser modificado para representar adequadamente a estrutura das redes de distribuição de energia e utilizam um algoritmo genético equipado com a ideia de vacinação previamente usada em (Keko et al., 2003) para aumentar seu desempenho.

Uma situação bem mais complexa surge no caso de roteamento dito *dinâmico* onde alguma característica ou parâmetro do problema não mais se mantém fixa: o ambiente operacional varia ao longo do tempo. Lau

& Wong (2005) propõem um arcabouço complexo (AISDR: AIS-Based Dynamic Routing) para o caso de expedição global e transbordo (*global shipment and transshipment*) de maneira a prover apoio ao processo de planejamento e avaliação de rotas de transporte a fim de eliminar limitações decorrentes da operação manual destes sistemas. A proposta do AISDR se apoia na seleção clonal, memória imunológica, regulação imune, reação imunológica cruzada e propriedades de redes imunológicas (Lau & Wong, 2005). Um paralelo entre cada característica do AISDR e seu correspondente no sistema imune biológico é apresentado nesta referência.

Finalmente, uma abordagem híbrida (neuro-imune) é apresentada por Masutti & de Castro (2009) para os casos do caixeiro viajante e de múltiplos caixeiros viajantes. O algoritmo RABNET-TSP (*Real-Valued Antibody Network to solve the TSP*), baseado nas ideias de mapas auto-organizáveis de Kohonen, utiliza o princípio de seleção clonal e maturação da afinidade para o desenvolvimento da rede neuronal.

4.3 Movimentação global de contêineres

Devido ao desequilíbrio da balança comercial entre países, contêineres vazios devem ser movidos de um local onde haja excesso deles para outro onde haja um déficit. Este reposicionamento deve ser feito da maneira mais econômica possível e Wong et al. (2010) consideram dois objetivos a serem minimizados: o custo total de reposicionamento (f_1) e a demanda não-satisfeita total (f_2) que podem ser escritos como na Equação (7):

$$C(Q) = \sum_{i=1}^{|S|} \sum_{j=1}^{|D|} \sum_{k=1}^{|V|} c_{ijk} q_{ijk} \quad \text{e} \quad U(Q) = \sum_{j=1}^{|D|} U_j \quad (7)$$

onde c_{ijk} é o custo de mover um contêiner vazio da região de excesso $i \in S$ para o local de deficit $j \in D$ através do navio $k \in V$ e q_{ijk} é a respectiva quantidade de contêineres a serem movidos. U_j é a demanda não-satisfeita no j -ésimo local de deficit e $|\cdot|$ denota cardinalidade do conjunto. Entretanto a minimização de C e U está sujeita a uma série de restrições sobre as variáveis de decisão q_{ijk} :

1. Restrições de disponibilidade: a quantidade de contêineres a serem movidos do i -ésimo local não pode exceder sua disponibilidade Q_i , como definido pela Equação (8):

$$\sum_{j=1}^{|D|} \sum_{k=1}^{|V|} q_{ijk} \leq Q_i \quad i = 1, 2, \dots, |S| \quad (8)$$

2. Restrições de capacidade: a quantidade total de contêineres transportados pelo k -ésimo navio não pode exceder sua capacidade Q_k , como definido pela Equação (9):

$$\sum_{i=1}^{|S|} \sum_{j=1}^{|D|} q_{ijk} \leq Q_k \quad (9)$$

3. Restrições de integralidade e não-negatividade das quantidades de contêineres, como definido pela Equação (10):

$$q_{ijk} \in \mathbb{Z} \quad \text{e} \quad q_{ijk} \geq 0 \quad (10)$$

Os problemas de otimização multiobjetivo buscam encontrar soluções ótimas que apresentem a melhor relação entre diversos objetivos, onde cada um deve ser minimizado ou maximizado. Sua formulação matemática pode ser escrita como na Equação (11):

$$\begin{aligned} &\text{minimizar } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \\ &\text{sujeito a } \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{S} \end{aligned} \quad (11)$$

onde $k(\geq 2)$ é o número de funções objetivo, \mathcal{S} é chamado conjunto ou região viável, \mathbf{f} é o vetor das funções objetivo com $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, \mathbf{x} é o vetor de decisão; se $\mathbf{x} \in \mathcal{S}$, então \mathbf{x} é uma solução viável. O conjunto $\mathcal{Z}(= f(\mathcal{S}))$ corresponde ao conjunto imagem da região viável, e cada elemento de \mathcal{Z} , chamado de objetivo, é escrito como $\mathbf{z} = (z_1, z_2, \dots, z_k)$, onde $z_i = f_i(\mathbf{x})$ para todo $i = 1, \dots, k$, são chamados valores dos objetivos.

Em geral, os objetivos são conflitantes e não-comensuráveis, impossibilitando que uma única solução otimize todos os objetivos. Assim, faz-se importante o conceito de otimalidade de Pareto, definido a seguir.

Dado $\mathbf{x} \in \mathcal{S}$ um vetor de decisão qualquer, diz-se que \mathbf{x} *domina* um outro vetor de decisão $\mathbf{x}' \in \mathcal{S}$, e escreve-se $\mathbf{x} \prec \mathbf{x}'$, se e somente se

$$f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \quad \forall i \in \{1, \dots, k\} \wedge \exists j \in \{1, \dots, k\} : f_j(\mathbf{x}) < f_j(\mathbf{x}')$$

onde k é o número de objetivos. O vetor de decisão \mathbf{x} é dito *não-dominado* com relação a um conjunto $\mathcal{S}' \subseteq \mathcal{S}$, se e somente se, não existir nenhum vetor em \mathcal{S}' que domina \mathbf{x} . O vetor objetivo $\mathbf{z} \in \mathbb{Z}$ é dito não-dominado, se o seu vetor de decisão \mathbf{x} correspondente é não-dominado. Um vetor de decisão \mathbf{x} é um ótimo de Pareto, se e somente se, \mathbf{x} é não-dominado com relação a \mathcal{S} . O vetor objetivo $\mathbf{z} \in \mathbb{Z}$ é dito ótimo de Pareto, se o seu vetor de decisão \mathbf{x} correspondente é um ótimo de Pareto. A melhoria de um vetor ótimo de Pareto em qualquer objetivo provoca a degradação em pelo menos algum outro objetivo.

O conjunto ótimo de Pareto contém todas as soluções ótimas de Pareto e o conjunto de vetores objetivos correspondente é denominado fronteira ótima de Pareto.

No caso de problemas com restrições, Wong et al. (2010) adotam o seguinte esquema de dominância: o anticorpo a_1 domina o anticorpo a_2 se uma das condições seguintes se verifica:

1. o anticorpo a_1 é factível (satisfaz às restrições) e a_2 infactível;
2. ambos são infactíveis mas a_1 viola menos restrições do que a_2 ;
3. ambos são infactíveis, violam a mesma quantidade de restrições mas a_1 domina a_2 segundo o conceito básico, envolvendo apenas f_1 e f_2 ;
4. ambas soluções são factíveis mas a_1 domina a_2 segundo o conceito básico.

Os principais passos do algoritmo podem então ser sumarizados como:

1. Inicialização da população com soluções geradas aleatoriamente;
2. Avaliação da afinidade de cada solução da população em relação ao seu nível de dominância (medido pelo número de soluções pelas quais ela é dominada) e de diversidade (dada pela medida de densidade populacional na região da solução);
3. Ordenamento da população de acordo com a dominância e, em caso de empate, de acordo com a densidade;
4. Divisão da população em três grupos: um subconjunto com os 30% de mais alta afinidade (SAA), os próximos 40% de afinidade média (SAM) e, finalmente, os de baixa afinidade (SAB);
5. Clonagem das soluções em SAA gerando o subconjunto SC;
6. Mutação das soluções em SAM gerando o subconjunto SM;
7. Supressão da população: as 10% piores soluções são eliminadas, gerando o subconjunto SS;
8. Recrutamento: geração aleatória de novas soluções, criando o subconjunto SR;
9. Reprodução: soluções em SR participam de crossover com as de SC gerando o conjunto SD
10. Iteração: a combinação dos conjuntos SC (clones), SM (mutações) e SD (diversificados) vão compor a próxima população.

Mais detalhes, bem como uma aplicação real deste algoritmo, podem ser encontrados em (Wong et al., 2010).

4.4 Outras aplicações

As aplicações de SIAs na Pesquisa Operacional, isoladamente ou hibridizadas com outras heurísticas, não se limitam, obviamente, a estas aqui destacadas. Muitas outras podem ser encontradas na literatura como, por exemplo, na programação de projetos sujeita a restrições de recursos (Agarwal et al., 2007; van Peteghem & Vanhoucke, 2009; van Peteghem & Vanhoucke, 2013; Mobini et al., 2011), na alocação de recursos numa rede de comunicações (Costanza et al., 2011) e na programação de estoque e produção (Wang et al., 2007).

5. Conclusões

Após uma apresentação geral dos principais aspectos do sistema imunológico, este capítulo apresentou um conjunto de técnicas computacionais imuno-inspiradas e finalizou descrevendo suas aplicações a problemas em Pesquisa Operacional.

6. Agradecimentos

Os autores agradecem o apoio financeiro do CNPq (Processos 308317/2009-2 e 300192/2012-6).

Referências

- Abbas, A.K.; Lichman, A.H. & Pober, J.S., *Imunologia Celular e Molecular*. 2a edição. Rio de Janeiro, RJ: Revinter, 1998.
- Ada, G.L. & Nossal, G.J.V., The clonal selection theory. *Scientific American*, 257(2):50–57, 1987.
- Agarwal, R.; Tiwari, M. & Mukherjee, S., Artificial immune system based approach for solving resource constraint project scheduling problem. *International Journal of Advanced Manufacturing Technology*, 34(5-6):584–593, 2007.
- Aickelin, U. & Cayzer, S., The danger theory and its application to artificial immune systems. In: *Proceedings of the 1st International Conference on Artificial Immune Systems*. p. 141–148, 2002.
- Bäck, T.; Fogel, D.B. & Michalewicz, Z. (Eds.), *Evolutionary Computation 2: Advanced Algorithms and Operations*. Bristol, UK: Institute of Physics Publishing, 2000.
- Bagheri, A.; Zandieh, M.; Mahdavi, I. & Yazdani, M., An artificial immune algorithm for the flexible job-shop scheduling problem. *Future Generation Computer Systems*, 26(4):533–541, 2010.
- Bernardino, H.S. & Barbosa, H.J.C., Artificial immune systems for optimization. In: Chiong, R. (Ed.), *Nature-Inspired Algorithms for Optimisation*. Berlin, Germany: Springer, p. 389–411, 2009.
- Bernardino, H.S.; Barbosa, H.J.C. & Fonseca, L.G., Surrogate-assisted clonal selection algorithms for expensive optimization problems. *Evolutionary Intelligence*, 4:81–97, 2011.
- Burnet, F.M., A modification of Jerne's theory of antibody production using the concept of clonal selection. *Australian Journal of Science*, 20:67–69, 1957.
- Coello, C.A.C.; Rivera, D.C. & Cortés, N.C., Use of an artificial immune system for job shop scheduling. In: Timmis, J.; Bentley, P.J. & Hart, E. (Eds.), *Proceedings of the International Conference on Artificial Immune Systems*. Heidelberg, Germany: Springer-Verlag, v. 2787 de *Lecture Notes in Computer Science*, p. 1–10, 2003.
- Cormack, D.H., *HAM Histologia*. 9a edição. Rio de Janeiro, RJ: Guanabara Koogan, 1991.
- Costanza, J.; Cutello, V. & Pavone, M., A memetic immunological algorithm for resource allocation problem. In: Liò, P.; Nicosia, G. & Stibor, T. (Eds.), *Proceedings of the International Conference on Artificial Immune Systems*. Heidelberg, Germany: Springer-Verlag, v. 6825 de *Lecture Notes in Computer Science*, p. 308–320, 2011.
- Cutello, V.; Narzisi, G.; Nicosia, G. & Pavone, M., Clonal selection algorithms: A comparative case study using effective mutation potentials. In: Jacob, C.; Pilat, M.L.; Bentley, P.J. & Timmis, J.I. (Eds.), *Proceedings of the International Conference on Artificial Immune Systems*. Heidelberg, Germany: Springer-Verlag, v. 3627 de *Lecture Notes in Computer Science*, p. 13–28, 2005a.
- Cutello, V.; Narzisi, G.; Nicosia, G. & Pavone, M., An immunological algorithm for global numerical optimization. In: Talbi, E.G.; Liardet, P.; Collet, P.; Lutton, E. & Schoenauer, M. (Eds.), *Proceedings of the International Conference on Artificial Evolution*. Heidelberg, Germany: Springer-Verlag, v. 3871 de *Lecture Notes in Computer Science*, p. 284–295, 2005b.
- Cutello, V. & Nicosia, G., An immunological approach to combinatorial optimization problems. In: Garijo, F.J.; Riquelme, J.C. & Toro, M. (Eds.), *Proceedings of the Ibero-American Conference on Artificial Intelligence*. Heidelberg, Germany: Springer-Verlag, v. 2527 de *Lecture Notes in Computer Science*, p. 361–370, 2002.
- Cutello, V.; Nicosia, G. & Pavone, M., A hybrid immune algorithm with information gain for the graph coloring problem. In: Cantú-Paz, E. (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference*. Heidelberg, Germany: Springer-Verlag, *Lecture Notes in Computer Science*, 2003.
- Cutello, V.; Nicosia, G. & Pavone, M., Exploring the capability of immune algorithms: A characterization of hypermutation operators. In: Nicosia, G.; Cutello, V.; Bentley, P.J. & Timmis, J. (Eds.), *Proceedings of the International Conference on Artificial Immune Systems*. Heidelberg, Germany: Springer-Verlag, v. 3239 de *Lecture Notes in Computer Science*, p. 263–276, 2004.
- Cutello, V.; Nicosia, G. & Pavone, M., An immune algorithm with stochastic aging and Kullback entropy for the chromatic number problem. *Journal of Combinatorial Optimization*, 14(1):9–33, 2007.
- de Castro, L.N., *Engenharia Imunológica: Desenvolvimento e Aplicações de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais*. Tese de doutorado, Universidade Estadual de Campinas, Campinas, SP, 2001.
- de Castro, L.N. & Timmis, J., An artificial immune network for multimodal function optimization. In: *Proceedings of the Congress on Evolutionary Computation*. Piscataway, USA: IEEE Press, v. 1, p. 699–704, 2002a.
- de Castro, L.N. & Timmis, J., *Artificial Immune Systems: A New Computational Intelligence Approach*. London, UK: Springer, 2002b.
- de Castro, L.N. & Timmis, J., *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Berlin, Germany: Springer-Verlag, 2002c.
- de Castro, L.N. & von Zuben, F.J., The clonal selection algorithm with engineering applications. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. p. 37, 2000.
- de Castro, L.N. & von Zuben, F.J., aiNet: An artificial immune network for data analysis. In: Abbass, H.A.; Sarker, R.A. & Newton, C.S. (Eds.), *Data Mining: A Heuristic Approach*. Hershey, USA: Idea Group Publishing, p. 231–259, 2001.

- de Castro, L.N. & von Zuben, F.J., Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, 2002.
- Gao, J., A novel artificial immune system for solving multiobjective scheduling problems subject to special process constraint. *Computers & Industrial Engineering*, 58(4):602–609, 2010.
- Garrett, S.M., Parameter-free, adaptive clonal selection. In: *Proceedings of the Congress on Evolutionary Computation*. Piscataway, USA: IEEE Press, v. 1, p. 1052–1058, 2004.
- Goldberg, D.E. & Richardson, J., Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. Mahwah, USA: Lawrence Erlbaum Associates, p. 41–49, 1987.
- Golub, E.S., Is the function of the immune system only to protect? In: *Theoretical and Experimental Insights into Immunology*. Berlin, Germany, v. 66 de *NATO ASI Series*, p. 15–26, 1992.
- Greensmith, J.; Aickelin, U. & Cayzer, S., Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In: Jacob, C.; Pilat, M.L.; Bentley, P.J. & Timmis, J.I. (Eds.), *Proceedings of the 4th International Conference on Artificial Immune Systems*. Heidelberg, Germany: Springer-Verlag, v. 3627 de *Lecture Notes in Computer Science*, p. 153–167, 2005.
- Hart, E. & Ross, P., The evolution and analysis of potential antibody library for use in job-shop scheduling. In: Corne, D.; Dorigo, M.; Glover, F.; Dasgupta, D.; Moscato, P.; Poli, R. & Price, K.V. (Eds.), *New ideas in optimization*. Maidenhead, UK: McGraw-Hill, p. 185–202, 1999.
- Hart, E.; Ross, P. & Nelson, J., Producing robust schedules via an artificial immune system. In: *Proceedings of the International Conference on Computational Intelligence*. Piscataway, USA: IEEE Press, p. 464–469, 1998.
- Hofmeyr, S.A. & Forrest, S., Architecture for an artificial immune system. *Evolutionary Computation*, 8(4):443–473, 2000.
- Janeway, C.A.; Travers, P.; Walport, M. & Shlomchik, M., *Immunobiologia: O sistema imune na saúde e na doença*. 5a edição. Porto Alegre, RS: Artes Médicas, 2001.
- Keko, H.; Skok, M. & Skrllec, D., Artificial immune systems in solving routing problems. In: *Proceedings of the International Conference on Computer As A Tool*. v. 1, p. 62–66, 2003.
- Keko, H.; Skok, M. & Skrllec, D., Solving the distribution network routing problem with artificial immune systems. In: *Proceedings of the Mediterranean Electrotechnical Conference*. v. 3, p. 959–962, 2004.
- Kelsey, J. & Timmis, J., Immune inspired somatic contiguous hypermutation for function optimisation. In: Cantú-Paz, E. (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference*. Heidelberg, Germany: Springer-Verlag, v. 2723 de *Lecture Notes in Computer Science*, 2003.
- Lau, H. & Wong, E., An AIS-based dynamic routing (AISDR) framework. In: Jacob, C.; Pilat, M.; Bentley, P. & Timmis, J. (Eds.), *Proceedings of the International Conference on Artificial Immune Systems*. Heidelberg, Germany: Springer-Verlag, v. 3627 de *Lecture Notes in Computer Science*, p. 56–71, 2005.
- Lee, Y.C. & Zomaya, A.Y., Immune system support for scheduling. In: Prokopenko, M. (Ed.), *Advances in Applied Self-organizing Systems*. Springer London, Advanced Information and Knowledge Processing, p. 247–270, 2008.
- Lin, S. & Kernighan, B.W., An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.
- Luh, G.C. & Chueh, C.H., A multi-modal immune algorithm for the job-shop scheduling problem. *Information Sciences*, 179(10):1516–1532, 2009.
- Ma, J.; Gao, L. & Shi, G., An improved immune clonal selection algorithm and its applications for VRP. In: *Proceedings of the International Conference on Automation and Logistics*. p. 2097–2100, 2009.
- Mahfoud, S.W., *Niching methods for genetic algorithms*. PhD Thesis, University of Illinois at Urbana-Champaign, Champaign, USA, 1995.
- Masutti, T.A. & de Castro, L.N., Neuro-immune approach to solve routing problems. *Neurocomputing*, 72(10-12):2189–2197, 2009.
- Matzinger, P., Tolerance, danger, and the extended family. *Annual review of immunology*, 12(1):991–1045, 1994.
- Mobini, M.; Mobini, Z. & Rabbani, M., An artificial immune algorithm for the project scheduling problem under resource constraints. *Applied Soft Computing*, 11(2):1975–1982, 2011.
- Pavone, M.; Narzisi, G. & Nicosia, G., Clonal selection: an immunological algorithm for global optimization over continuous spaces. *Journal of Global Optimization*, 53(4):769–808, 2012.
- van Peteghem, V. & Vanhoucke, M., An artificial immune system algorithm for the resource availability cost problem. *Flexible Services and Manufacturing Journal*, 25(1-2):122–144, 2013.
- Tizard, I., *Introdução à Imunologia Veterinária*. 2a edição. São Paulo, SP: Roca, 1985.
- van Peteghem, V. & Vanhoucke, M., An artificial immune system for the multi-mode resource-constrained project scheduling problem. In: Cotta, C. & Cowling, P. (Eds.), *Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization*. Heidelberg, Germany: Springer-Verlag, v. 5482 de *Lecture Notes in Computer Science*, p. 85–96, 2009.

- Wang, X.P.; Zhao, H. & Shang, H.Y., Simulation and optimization of a production/inventory system based on immune evolutionary algorithm. In: *Proceedings International Conference on Management Science and Engineering*. Piscataway, USA: IEEE Press, p. 796–801, 2007.
- Wong, E.; Lau, H. & Mak, K., Immunity-based evolutionary algorithm for optimal global container repositioning in liner shipping. *OR Spectrum*, 32:739–763, 2010.
- Yamada, T. & Nakano, R., Job-shop scheduling. In: *Genetic Algorithms in Engineering Systems*. Stevenage, UK: The Institution of Electrical Engineers, Control engineering, p. 134–160, 1997.

Notas Biográficas

Grazziela Patrocínio Figueredo é graduada em Informática (Universidade Federal de Juiz de Fora, 2002), mestre em Engenharia de Sistemas e Computação e doutor em Computação de Alto Desempenho (COPPE/UFRJ, 2004 e 2008, respectivamente). Atualmente trabalha como pesquisadora na Universidade de Nottingham. Tem experiência na área de Ciência da Computação, com ênfase em Inteligência Artificial, atuando principalmente em modelagem e simulação computacional, sistemas imunes artificiais e mineração de dados.

Heder Soares Bernardino é graduado em Ciência da Computação e mestre em Modelagem Computacional (Universidade Federal de Juiz de Fora, 2006 e 2008, respectivamente). Atualmente é aluno de doutorado no programa de pós-graduação do Laboratório Nacional de Computação Científica (LNCC/MCTI). Interessa-se em meta-heurísticas, em especial para a descoberta de conhecimento, inferência de modelos e geração automática de projetos estruturais, bem como em computação de alto desempenho. Atualmente é professor do departamento de Ciência da Computação da UFJF.

Helio José Corrêa Barbosa é graduado em Engenharia Civil (Universidade Federal de Juiz de Fora, 1974), mestre e doutor em Engenharia Civil (COPPE/UFRJ, 1978 e 1986, respectivamente). Atualmente é Tecnologista Senior no Laboratório Nacional de Computação Científica e Professor Associado do Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora. Tem interesse no desenho e aplicação de meta-heurísticas em engenharia e biologia.