

CAPÍTULO 3

ALGORITMOS MEMÉTICOS

3.1 Introdução

Os cinco anos em que Charles Darwin atuou como biólogo em sua viagem a bordo do navio Beagle, permitiram que ele coletasse uma grande quantidade de informação e material biológico. Toda essa massa de dados, em conjunto com as suas observações, certamente contribuíram para a formulação das suas idéias sobre os mecanismos da evolução (SILVA & SASSON, 1996). Pode-se inferir que Darwin, ao elaborar a sua Teoria da Evolução, também expôs suas idéias (um conjunto de observações, conhecimento adquirido, genialidade e conclusões) a um processo de seleção. Ele construiu uma “pré-história” sempre confrontando as conclusões obtidas com os fatos observados e com os dados coletados. Este método de trabalho fez com que as características em comum fossem mantidas, e as outras, descartadas. Na verdade, Darwin trabalhava com uma população de idéias, ou em outras palavras, “indivíduos de concepção mental” que passavam por um processo de busca pelo melhor entendimento dos dados coletados. É também provável que, com o conhecimento adquirido ao longo do tempo, ele tenha aperfeiçoado seus mecanismos de busca e seus critérios de avaliação. Este processo de busca pela melhor resposta somente pode ser concebido dentro dos limites do campo do conhecimento, na forma de um processo iterativo aplicado por um indivíduo para a melhoria continuada de suas idéias. Este processo sofisticado de refinamento do conhecimento se justifica no caso de problemas que envolvem um grande número de possibilidades de interpretação e formalização.

3.2 Memes

No campo da computação evolutiva, o refinamento do conhecimento pode ser incorporado como uma etapa de apoio ao processo evolutivo. MOSCATO & NORMAN (1992) introduziram o termo “algoritmo memético” para descrever um processo evolutivo que possua uma busca local como parte decisiva na evolução. Essa busca pode ser caracterizada

como sendo um refinamento local dentro de um espaço de busca. O termo *meme* foi idealizado por DAWKINS (1976) como sendo uma unidade de informação que se reproduz durante um processo argumentativo e de transmissão de conhecimento (RADCLIFFE & SURRY, 1994). Portanto, pode-se dizer que, enquanto o algoritmo memético está relacionado com a evolução cultural, o algoritmo genético está baseado na evolução biológica dos indivíduos.

Uma diferença marcante entre genes e *memes* está no processo de transmissão aos seus descendentes. Quando o *meme* é transmitido, ele será adaptado pela entidade que o recebe com base no seu conhecimento e para melhor atender suas necessidades. Quanto aos genes, no processo de evolução eles são transmitidos de uma maneira tal que o descendente gerado vai herdar muitas habilidades e características presentes em seus progenitores. Dessa maneira, o algoritmo genético é inspirado na tentativa de emulação computacional da evolução biológica e o algoritmo memético tenta fazer o mesmo em relação à evolução cultural.

É interessante lembrar que na evolução biológica a informação está codificada nos genes por uma seqüência de nucleotídeos, de modo que a transmissão é influenciada pela presença de mutação, recombinação e pela seleção natural em relação aos indivíduos geneticamente melhor adaptados. Na evolução cultural, a informação envolvida está nos *memes*, de modo que as alterações surgem pela combinação, criação e reorganização das representações mentais (conscientes ou não) e pela possível ineficácia dos mecanismos de transmissão de informação. A replicação (fenótipo) ocorre quando essas representações mentais são transformadas em ações passíveis de imitação ou expressas através de alguma linguagem. A incorporação dessa nova informação por parte de algum indivíduo certamente alterará a pressão seletiva e a influência vinculada às limitações impostas pelo ambiente.

3.3 Algoritmos Meméticos

De acordo com MOSCATO (1999), o uso genérico da denominação de Algoritmo Memético é feito para a identificação de uma classe de meta-heurísticas, constituindo um dos procedimentos de maior sucesso para problemas de otimização combinatória.

A característica principal, presente em muitas implementações que utilizam algoritmos meméticos, é o uso de processos de busca dedicados. Esses processos pretendem

utilizar toda a informação disponível sobre o problema, de modo que este conhecimento seja incorporado sob a forma de heurísticas, técnicas de busca local, operadores especializados de recombinação e muitas outras maneiras.

Em essência, os algoritmos meméticos podem ser interpretados como um conjunto de estratégias que implementam a competição e a cooperação entre diferentes mecanismos de otimização (MOSCATO & NORMAN, 1992). Sendo assim, o sucesso obtido pode ser explicado como sendo uma consequência direta da sinergia dos diferentes processos de busca utilizados.

A idéia geral dos algoritmos meméticos é a utilização dos operadores evolutivos que determinam regiões promissoras no espaço de busca, combinados com busca local nestas regiões (este processo tem sido aplicado com sucesso em vários problemas de otimização – MERZ & FREISLEBEN, 1999). Também pode-se dizer que algoritmos meméticos correspondem à união de um método de busca global e uma heurística local aplicada a cada indivíduo, de modo que um algoritmo memético é, na verdade, um tipo especial de *hill-climbing*.

Conforme MERZ & FREISLEBEN (1999), em um ambiente que emprega algoritmo memético, os operadores de recombinação e mutação agem como estratégias de diversificação. Os indivíduos da população podem estar localizados em uma região do espaço de busca contendo um ótimo local, chamada base de atração do ótimo local. Utilizando a informação contida na população, novos pontos de partida podem ser descobertos após a busca local. Os operadores de recombinação e mutação podem gerar indivíduos da população que estejam localizados em bases de atração de ótimos locais ainda não explorados, de modo que um novo pico deva ser alcançado (maximização) ou um vale deva ser explorado (minimização). Utilizando o conceito de superfície de adaptação (*fitness*), a Figura 3.1 ilustra estes eventos, no caso da maximização. Após a recombinação, o filho gerado pode possuir um *fitness* baixo, mas um grande potencial para crescimento, de modo que uma busca local pode levar o descendente a assumir um valor de *fitness* elevado. A mutação, por sua vez, pode levar a um pequeno aumento (ou decréscimo) do *fitness*, por representar uma perturbação local junto à representação do indivíduo.

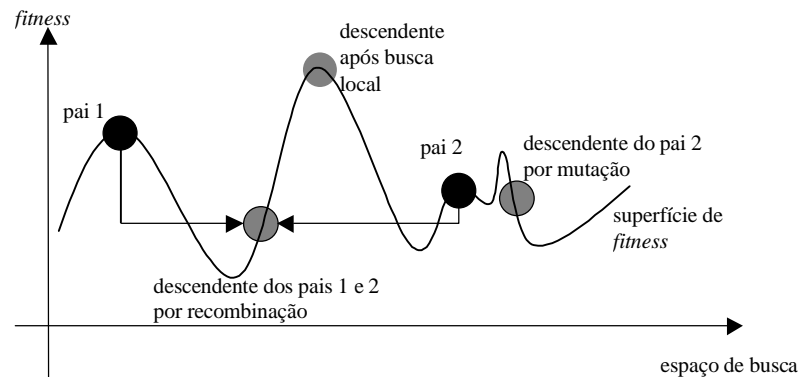


Figura 3.1: Operadores de recombinação e mutação agindo como estratégias de diversificação junto a algoritmos meméticos.

Na presença de restrições, os operadores genéticos de recombinação e mutação podem produzir soluções que estão fora da região factível do espaço de busca. Entretanto, um algoritmo de reparação (factibilização) pode ser elaborado para remeter o descendente gerado para uma região factível (RADCLIFFE & SURRY, 1994). Embora também pudesse ser caracterizado como um processo de busca local, este algoritmo de busca será denominado algoritmo de reparação, a ser descrito no capítulo 4. A razão para a distinção entre uma busca local realizada por um algoritmo memético e a busca realizada por um algoritmo de reparação é que este último não investe no aumento da qualidade da solução, mas apenas na garantia de sua factibilidade.

No caso de problemas de otimização irrestritos, a geração da população inicial é realizada em duas etapas. A primeira, consiste na geração de indivíduos que procuram explorar amplamente o espaço de busca, caracterizando-os como soluções candidatas. Na segunda etapa, procedimentos de busca local são implementados para a obtenção de um ótimo associado à região do espaço em que se encontra cada indivíduo. Esse mesmo processo de busca é empregado após a aplicação de qualquer operador genético (MERZ & FREISLEBEN, 1999), procurando remeter o descendente gerado a um ótimo local.

No caso de problemas de otimização com restrições, já na geração da população inicial podem surgir indivíduos infactíveis, não porque possuam um genótipo mal formado ou porque o algoritmo de solução utilizado contenha erros, mas sim por violarem restrições

impostas ao problema. Há, portanto, a necessidade de algum dispositivo de seleção que seja capaz de discernir entre indivíduos factíveis ou não. Em algumas aplicações, como é o caso deste trabalho, é possível implementar algoritmos de reparação para levar indivíduos à factibilidade (eventualmente à factibilidade mais próxima). Uma vez tendo essa necessidade atendida, é uma consequência natural o emprego de uma busca local sobre os indivíduos factíveis, fazendo com que eles possam atingir um ótimo local dentro da região factível. Portanto, resulta como abordagem de solução um algoritmo memético (ou algoritmo genético associado a mecanismos de busca local) aliado a etapas de reparação, tanto para conduzir os indivíduos à factibilidade como para aumentar o seu *fitness*.

3.4 Formalização do Processo de Busca Local

AARTS & VERHOEVEN (1997) consideram a busca local como sendo uma aproximação geral para problemas de otimização combinatória baseados na exploração de vizinhanças (para maiores detalhes veja AARTS & LENSTRA, 1997).

De maneira genérica, um algoritmo de busca local começa com um indivíduo $s_0 \in S$ e tenta continuamente encontrar melhores elementos dentre os vizinhos. Em outras palavras, a finalidade será remeter esse indivíduo para um local onde a função de *fitness* tenha um valor melhor que o atual (Figura 3.1). Essa busca deverá ser realizada até que uma condição de parada seja satisfeita, sendo que essa condição deve ser atendida sempre que o processo de busca não tenha mais a capacidade de melhorar a solução atual. Observe o algoritmo representativo de um processo de busca na Figura 3.2.

Procedimento busca local**início****repita**final \leftarrow verdade**para** $i \leftarrow 1$ **até** número de indivíduos **faça****início**elemento \leftarrow indivíduo(i)

aplicar busca local (elemento)

se *fitness* é melhor **então** final \leftarrow falso**fim****até final****fim**

Figura 3.2: Algoritmo genérico representativo de uma busca local.

MOSCATO (1999) apresenta uma definição formal de busca local que será apresentada a seguir.

Um problema computacional P tem domínio de entrada I_p , com $x \in I_p$, podendo ser estabelecido um conjunto $ans_p(x)$ de respostas correspondentes. Entretanto, é preciso garantir que exista um subconjunto $sol_p(x) \subseteq ans_p(x)$ que identifica as soluções factíveis de P . Um algoritmo soluciona um problema P se, para a entrada $x \in I_p$, apresentar como saída qualquer $y \in sol_p(x)$ – solução factível – ou, no caso de $sol_p(x) = \{ \}$, indicando que não existe y . A otimização combinatória é um tipo especial de problema de busca, em que cada $x \in I_p$ tem um conjunto $sol_p(x)$ de cardinalidade finita e cada solução $y \in sol_p(x)$ tem um valor de *fitness* $m_p(y, x)$. A busca, nesse tipo de problema, será responsável por encontrar uma solução factível $y^* \in sol_p(x)$ que maximize o *fitness* $m_p(y, x)$.