NSGA-II
Explicação geral do algoritmo evolutivo

GUSTAVO FERNANDES DE ALMEIDA
gsutavo@outlook.com
Universidade de Brasília
November 22, 2015

Abstract

O objetivo desse documento é explicar como o algoritmo de otimização multiobjetivo NSGA-II[1] funciona. Em especial busca-se aprofundar em como sua implementação é feita.

Keywords: NSGA-II , elitista , algoritmo genético , otimização multiobjetivo

Introdução

O NSGA-II é o aprimoramento do algoritmo genético de ordenação não-dominada (NSGA) e busca solucionar três problemas que o seu anterior apresentava:

- Complexidade computacional alta da ordenação não-dominada;
- Falta de elitismo;
- Necessidade de especificar o parâmetro de compartilhamento σ_{share}.

O prejuízo da alta complexidade computacional ($O(MN^3)$), M é o número de objetivos e N o tamanho da população, torna o NSGA muito caro para populações largas, essa complexidade tem base no processo de ordenação realizado em cada geração do algoritmo.

Elitismo em um algoritmo evolutivo discrimina soluções em melhores e piores dependendo de sua performance. As soluções que apresentam resultados melhores são priorizadas durante as outras fases do algoritmo,

em especial durante a seleção para formar a *mating pool*[1]. Elitismo trás boas contribuições a um algoritmo genético como aumentar sua performance e prevenir a perda de uma boa solução depois de ser encontrada.

Mecanismos tradicionais para garantir a diversidade em uma população afim de obter uma variedade de soluções equivalentes tem se baseado no conceito de compartilhamento. Um problema dessa abordagem é a necessidade de um parâmetro de compartilhamento que deve ser estabelecido. Um algoritmo sem esse requisito seria mais desejável.

Sobre o algoritmo

Para explicar os pilares que compõe o NSGA-II temos que passar por três etapas: como é feita a *Fast Nondominated Sorting Approach*[2], como preservar a diversidade das soluções e como o *main loop* é feito.

Fast Nondominated Sorting Approach

Essa abordagem busca resolver o problema da alta complexidade computacional que o NSGA tinha. Utilizando a maneira proposta a seguir $O(MN^2)$ computações serão necessárias.

Antes de começar detalhar os passos do algoritmo é preciso compreender o conceito de dominação no contexto desse algoritmo. Uma solução x é dominante quando comparada a outra solução y quando todas os resultados das funções de otimização são melhores para x que para y. Se em qualquer uma das funções o resultado de y for melhor que x nenhuma das duas soluções é dominante ou dominada.

O primeiro passo é calcular duas entidade para cada solução:

- O contador de dominação n_p, ele representa o número de soluções que dominam a solução p;
- O conjunto de soluções S_p, que lista todas as soluções que p domina.

O primeiro *front* é determinado pela soluções que tiverem n_p igual a zero, ou seja, não são dominadas por nenhuma outra solução. Em seguida para cada solução p com $n_p = 0$ visitamos cada uma dos membros (q) de seu S_p e diminuímos o valor do seu n_p por uma contagem de um. Se o novo n_p de q for zero colocamos q numa nova lista Q. Q formará o segundo *front*. Repetimos esse processo com Q para determinar o terceiro *front*. Realizamos esse processo até que todos os *fronts* sejam definidos e todas as soluções façam parte de um *front*.

[1] *Mating pool*: estrutura de um algoritmo genético que representa o sub-grupo de uma população selecionado para reproduzir entre si e, portanto, passar seus genes adiante.

[2] Abordagem de ordenação não-dominada rápida em tradução livre

Preservação de Diversidade

A maneira com que o NSGA-II mantém a variedade das soluções é composta por duas estratégias: estimação de densidade e operador de *crowded-comparison*.

Para obter uma estimativa da densidade de soluções ao redor de uma solução particular calculamos a distância média de dois pontos aos lados da solução para cada objetivo. Essa distância, chamada de *crowdind distance*, é usada para estimar o perímetro do cuboide formado usando os vizinhos mais próximos como vértices.

Para calcular a *crowdind disteance* precisamos ordenar a população em ordem ascendente para cada uma das funções objetivo. Para cada função, as soluções nas extremidades (a de menor valor e a de maior) são as soluções de fronteira, suas distâncias são definidas como infinitas. As outras soluções intermediárias tem seus valores de distância calculado como a diferença absoluta normalizada dos valores das funções das suas duas soluções adjacentes. Esse cálculo é feito em todas funções de objetivo para cada uma das soluções. A *crownding distance* total é a soma de cada um dos valores de distância para cada função objetivo. As funções objetivo devem ser normalizadas antes de calcular a *crowding distance*.

Quando cada solução tem seu valor de *crowding distance* calculado é possível avaliar as soluções do ponto de vista de diversidade. *Crowing distance* pequena significa que a solução esta numa área densa de soluções parecidas enquanto um valor alto é indicativo de uma solução mais alternativa.

O operador de *crowded-comparison* ($\prec_n$) guia o processo de seleção em vários estágios do algoritmo afim de uniformemente espalhar o *front* de Pareto ótimo.

Para um cada indivíduo i de uma população sendo i_{rank} o rank de não-dominação e $i_{distance}$ o valor da *crowding distance* temos:

$$\text{i} \prec_n j \; se \; (i_{rank} < j_{rank})$$
$$\text{OU } ((\text{i}_{rank} = j_{rank}) E (i_{distance} > j_{distance}))$$

Dessa maneira respeitamos o elitismo enquanto mantemos uma variedade alta. Uma solução de rank inferior não é escolhida enquanto os *fronts* melhores ainda não estiverem esgotados e soluções diferentes são priorizadas.

Main Loop

O *main loop* define a iteração do algoritmo. Inicialmente uma população P_0 é criada aleatoriamente. A população é ordenada em relação ao seu fator de não-dominação. Esse fator é o level de não-dominação, ele representa o valor de *fitness* ou rank. Level 1 é o melhor, 2 é o segundo e assim por

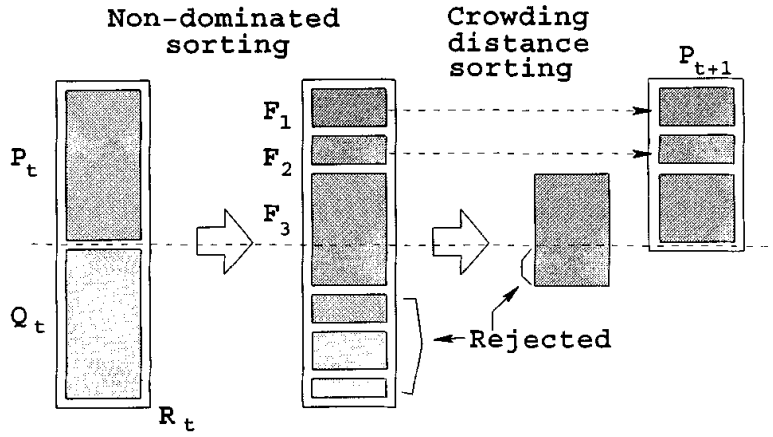

Figure 1: Figura representativa do processo do NSGA-II retirada do artigo *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*[1]

diante. Utilizando técnicas de seleção por torneio binário, recombinação e operadores de mutação criamos a população descendente Q_0 de tamanho N. A inicialização do algoritmo é diferente das outras iterações.

O laço principal segue a sequência a seguir:

- A população $R_t = P_t \cup Q_t$ de tamanho 2N é formada;
- A população R_t é ordenada de acordo com o valor de não-dominação;
- O conjunto de soluções do primeiro *front* (F_1), as não-dominadas por nenhuma outra solução, são enfatizadas mais que qualquer outro grupo. Se F_1 for menor que N todos seus membros irão formar a nova população P_{t+1};
- Outros membros de P_{t+1} são selecionados dos outros *fronts* pela ordem de seu ranking até completar N membros;
- Caso existam mais membros no *fronts* do que vagas em P_{t+1} o operador de *crowding-comparison* ($\prec_n$) define os indivídos o *front* que completarão as vagas remancentes;
- A nova população P_{t+1} de tamanho N passa por seleção, *crossover* e mutação para criar uma nova Q_{t+1} de tamannho N;
- Outra iteração é feita até o número de gerações predeterminado é atingido.

A figura 1 ilustra o processo que o algoritmo NSGA-II define.

Conclusão

O NSGA-II atinge todas os objetivos que se propõe, tornando-o uma boa alternativa ao seu antecessor.

Em especial, ele retira a necessidade de um parâmetro de compartilhamento introduzindo o procedimento de *crowding comparison* além de aplicar a noção de elitismo e ser computacionalmente mais eficiente o NSGA.

References

1. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.