Universidade de Pernambuco (UPE)
Escola Politécnica de Pernambuco (POLI)
Programa de Pós-graduação em Engenharia da Computação (PPGEC)

Relatório da Prática de Algoritmos de Otimização Multi-Objetivos
MOPSO-CDR

Aluno: Carlos Henrique Maciel Sobral Timoteo

Professor: Dr. Carmello Bastos Filho

17 de Setembro de 2013

Lista de Figuras

Sumário

Capítulo 1

Introdução

A motivação deste estudo é realizar a aplicação de uma técnica de algoritmo de otimização multi-objetivos, mais especificamente o *MOPSO-CDR* (do inglês *Multiple Objective Particle Swarm Optimization - Crowding Distance and Roulette Whell*, Otimização por Enxame de Partículas Multi-objetivos utilizando crowding distance e roleta), e apresentar os resultados práticos da aplicação da técnica.

A prática consiste em responder a seguinte pergunta: Supondo que temos uma base de dados de riscos em gerenciamento de projetos de software, quais os riscos que devem ser mitigados de modo que as catástrofes e os riscos com maior probabilidade não ocorram.

1.1 Algoritmo PSO

O PSO é um algoritmo inspirado no voo de aves em busca de alimentos. Foi idealizado por Kennedy e Eberhart em 1995 [1]. Cada partícula é uma solução candidata para um problema e, por meio de troca de informações, o conjunto de partículas (o enxame) realiza, iterativamente, uma busca para encontrar uma "boa"solução para o problema. A solução é representada pelo posição da partícula no espaço de busca. Não é possível afirmar que a solução final é a melhor pois seria necessário uma exploração total do espaço de busca e um estudo de convergência para o algoritmo implementado.

Como há movimentação de partículas pelo espaço de busca, é clara a presença de uma velocidade e de uma posição. O algoritmo original do PSO apresenta duas equações que definem a atualização da velocidade e da posição iterativamente. Estas equações são, respectivamente

$$\vec{v}[t+1] = \omega\vec{v}[t] + C_1 r_1(\vec{p}_{Best} - \vec{x}[t]) + C_2 r_2(\vec{g}_{Best} - \vec{x}[t]) \tag{1.1}$$

e

$$\vec{x}[t+1] = \vec{x}[t] + \vec{v}[t+1] \tag{1.2}$$

onde $\vec{v}$ e $\vec{x}$ são, respectivamente, a velocidade e a posição da partícula. C_1 e C_2 são números tal que $C_1, C_2 \in \mathbb{R}$ e representam a importância que a partícula dá a, respectivamente, informação cognitiva e a informação do enxame. r_1 e r_2 são números uniformemente distribuídos no intervalo $[0, 1]$ e ω representa a inércia da partícula.

1.2 Algoritmo MOPSO-CDR

O MOPSO-CDR foi proposto por Santana *et al.* [2] em 2009. Ele foi inspirado no MOPSO-CDLS e incorpora uma seleção com roleta baseado na crowding distance para eleger o líder social (*gBest*) e evitar um número excessivo de soluções não-dominadas no arquivo externo. A solução com menor crowding distance tem mais chance de ser selecionada como um líder social.

1.3 Base de Dados - PERIL 2008

A base de dados PERIL 2008, é uma base de dados desenvolvida por Tom Kendrick [3], resultado de cinco anos de pesquisa e coleta de informações de riscos que ocorreram em diversas projetos de tecnologia da informação. Nela encontramos 649 registros de riscos e o impacto expresso em tempo devido a sua ocorrência. O autor também realizou uma classificação dos riscos em dois grandes grupos: riscos comuns e catástrofes (*black swam*), além de categorizar os riscos.

Capítulo 2

Planejamento

O risco pode ser compreendido matematicamente como o produto da probabilidade de ocorrência de um risco e o impacto. Já, catástrofe é o impacto somado com logaritmo neperiano da probabilidade de ocorrência do risco.

2.1 Função para otimização

2.1.1 Risco

A função Risco é definida tal que

$$f(P, i) = Pi \tag{2.1}$$

onde $P \in [0; 1]$ indica a probabilidade de ocorrência de um risco e i indica o impacto esperado quando o risco ocorrer. A Figura 2.1 apresenta o formato da função quando $i = 20$

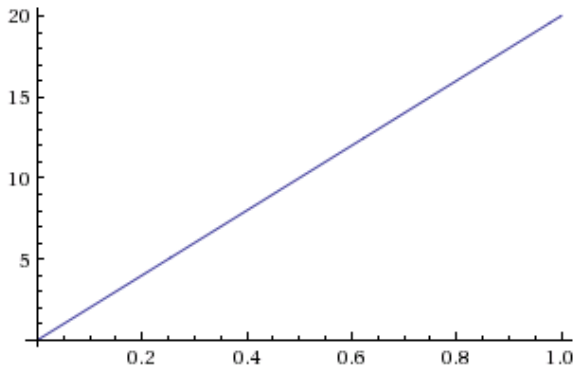

Figura 2.1: Função Risco para i = 20.

2.1.2 Catástrofe

A função Catástrofe é definida tal que

$$g(P, i) = i + \ln(P) \tag{2.2}$$

onde $P \in [0; 1]$ indica a probabilidade de ocorrência de um risco e i indica o impacto esperado quando o risco ocorrer. A Figura 2.2 apresenta o formato da função quando $i = 20$

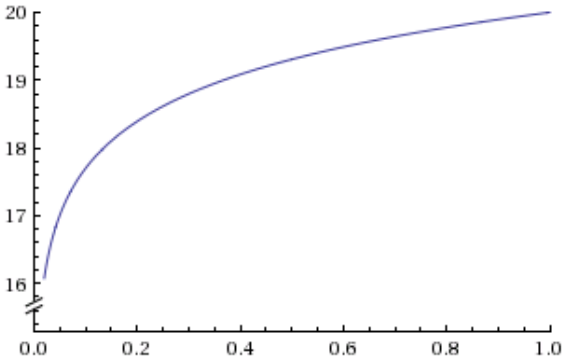

Figura 2.2: Função Catástrofe para i = 20.

Sobrepondo as duas funções, o objetivo é encontrar o *pareto front* ou o conjunto de soluções dominantes para as duas funções acima, levando em consideração que existe um custo máximo a ser satisfeito. A Figura 2.3 apresenta as funções 2.1 e 2.2 sobrepostas.

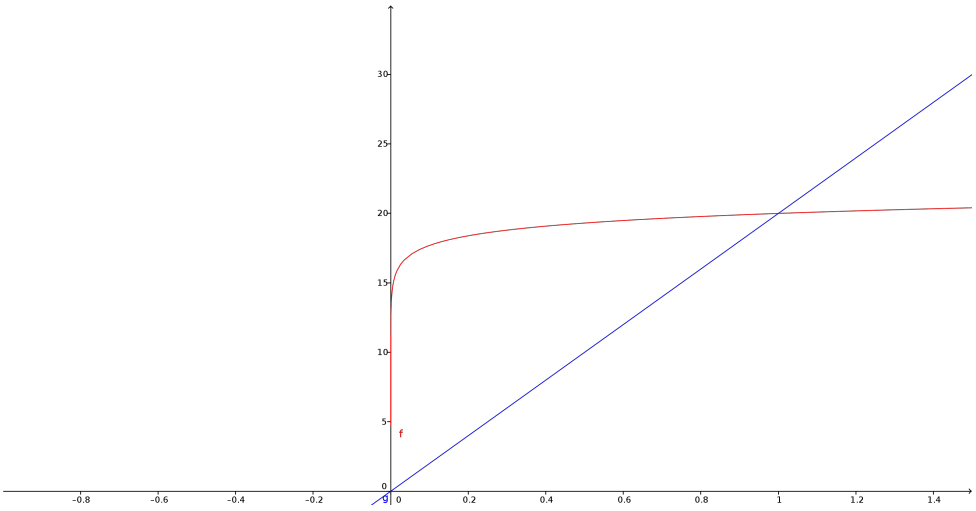

Figura 2.3: Funções Risco e Catástrofe Sobrepostas para i = 20.

Capítulo 3

Operação

O experimento consiste de três fases. A primeira é realizar o pré-processamento da base de dados, para o estudo os dados foram normalizados segundo a distribuição gamma. Por fim, foram separados em três conjuntos: conjunto de treinamento, conjunto de validação cruzada e conjunto de testes.

A segunda fase consiste em realizar a previsão dos dados de testes utilizando uma rede neural MLP, e utilizar os impactos esperados para a fase seguinte.

A terceira fase é identificar quais os riscos que devem ser priorizados após a otimização multi-objetivo realizada com o algoritmo MOPSO-CDR [2] utilizando como dados de entrada o conjunto de testes após a previsão. O algoritmo programado foi escrito na linguagem de programação Java.

Como saída do programa, tem-se o melhor resultado já visitado pelo enxame para cada 1000 iterações. A configuração do computador utilizado para realizar os experimentos é: Intel Core 2 Duo 2,00GHz, 2GB de memória RAM, sistema operacional Windows 7.

Capítulo 4

Resultados

Este capítulo apresenta os resultados obtidos para o problema investigado. Nesse caso, utilizamos a função de *fitness* Risco(F1) e Catástrofe (F2) e exibimos os resultados a seguir.

4.1 Resultados obtidos

As figuras a seguir são o arquivo externo e as melhores soluções representadas pela posição das partículas.

A Figura 4.1 apresenta so gráficos na iteração 20000.

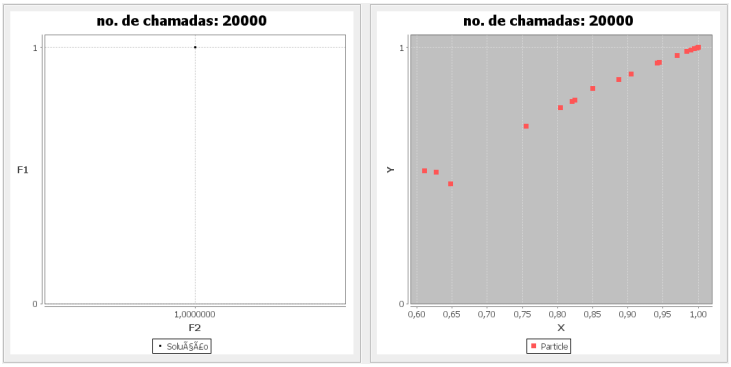

Figura 4.1: Gráfico para iteração 20000

A Figura 4.2 apresenta so gráficos na iteração 20000.

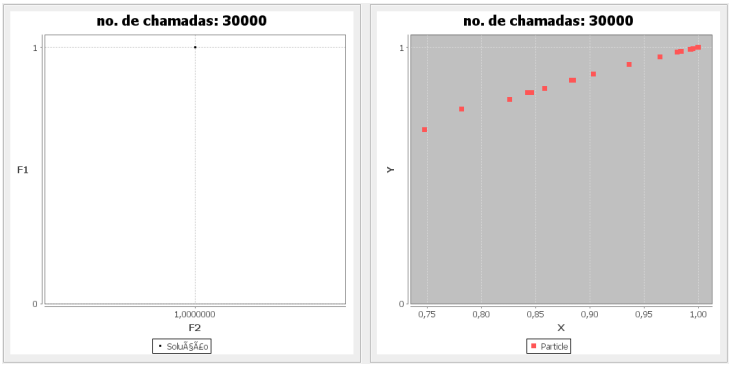

Figura 4.2: Gráfico para iteração 30000

Capítulo 5

Conclusão

O algoritmo MOPSO-CDR é recente e é comumente utilizado para otimizar funções clássicas como ZDT-1, ZDT-2, Sphere-2D. Para este trabalho realizou-se um estudo de aplicação do MOPSO-CDR para um problema de priorização de riscos.

Observou-se que o algoritmo foi capaz de fazer as partículas convergirem para o pareto ótimo representado pelo ponto de coordenadas (1.0, 1.0). No entanto, ele necessita de adaptações para que as partículas não convirgam para o ponto ótimo. Porque na prática, não temos recursos suficientes para contigenciar os riscos/catástrofes que apresentam os maiores valores agregados.

Portanto, soluções sub-ótimas como as apresentadas nas iterações 20.000 e 30.000, apresentadas nas Figuras 4.1 e 4.2, em algumas execuções mostram o resultado prático esperado.

Partindo desses resultados preliminares, sugere-se um estudo mais aprofundado para obtenção de melhores resultados práticos.

Referências Bibliográficas

[1] R. Kennedy, J.; Eberhart. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1995.

[2] R.A. Santana, M.R. Pontes, and C.J.A. Bastos-Filho. A multiple objective particle swarm optimization approach using crowding distance and roulette wheel. *Intelligent Systems Design and Applications, International Conference on*, 0:237–242, 2009.

[3] Tom Kendrick. Identifying and managing project risk. 2, 2009.