

# Uso do irace para geração de parâmetros de um algoritmo de busca tabu utilizado na resolução do problema da mochila.

## O que é o Irace?

O pacote `irace` implementa um procedimento de varredura iterativa, que é uma extensão do F-race. O objetivo do uso do `irace` é na configuração automática da otimização e decisão de algoritmos, ou seja, encontrar a configuração mais apropriada de um algoritmo dado um conjunto de instâncias de um problema. Contudo ele pode também ser usado para configurar outros tipos de algoritmos quando a performance depende da configuração dos parâmetros.

## O problema da mochila

O problema consiste em preencher uma mochila com objetos de diferentes pesos e valores. O objetivo é que se preencha a mochila com o maior valor possível, não ultrapassando o seu peso máximo.

## Busca Tabu

O algoritmo desenvolvido foi a busca tabu (TS) é uma meta-heurística que orienta um procedimento de busca heurística local para explorar a solução espaço além da otimização local. O termo Tabu Search (TS) foi cunhado no mesmo artigo que introduziu o termo meta-heurística. A busca tabu é baseada na premissa de que a resolução de problemas, para qualificar como inteligente, deve incorporar memória adaptativa e exploração responsiva.

O código desenvolvido em python pode ser encontrado no arquivo ***IRaceBuscaTabuHesau.py*** que segue em anexo no trabalho.

## Configurações e preparação

Para realizar a otimização utilizou-se o computador Acer Intel Core i3-7100U(2.4GH) com memória de 4GB e 1000Gb de HDD. O sistema operacional utilizado foi o Linux Ubuntu 18.04. Editor de código utilizado foi o Visual Studio Code versão 1.60.

Figura 1 – Configuração do arquivo parameters.txt

# 1:	2:	3: 4:	5:
tempo	--tempo "	r	(0.05, 0.1)
T	--T "	r	(0.25, 0.6)

## Resultado do primeiro experimento

Figura 2 – Output das configurações

```
# Elitist race
# Elitist new instances: 1
# Elitist limit: 2
# nbIterations: 3
# minNbSurvival: 3
# nbParameters: 2
# seed: 768527117
# confidence level: 0.95
# budget: 5000
# mu: 5
# deterministic: FALSE

# 2021-09-15 10:18:39 -03: Iteration 1 of 3
# experimentsUsedSoFar: 0
# remainingBudget: 5000
# currentBudget: 1666
# nbConfigurations: 277
```

Figura 3 – Instâncias treinadas

	Instance	Alive	Best	Mean best	Exp so far	W time	rho	KenW	Qvar
x	1	277	1	-46.56900000	277	00:35:46	NA	NA	NA
x	2	277	1	-43.62450000	554	00:35:54	+1.00	1.00	0.0000
x	3	277	1	-43.78133333	831	00:35:43	+1.00	1.00	0.0000
x	4	277	1	-44.62950000	1108	00:35:36	+1.00	1.00	0.0000
=	5	277	1	-47.98020000	1385	00:35:39	+1.00	1.00	0.0000
=	6	277	1	-49.87900000	1662	00:35:59	+1.00	1.00	0.0000

Figura 4 – Resultados obtidos

```

Best-so-far configuration:          1    mean value:    -49.87900000
Description of the best-so-far configuration:
  .ID.  tempo    T .PARENT.
  1    1 0.0898 0.403      NA

# 2021-09-15 13:53:20 -03: Elite configurations
#(first number is the configuration ID;
#listed from best to worst according to the sum of ranks):
  tempo    T
  1 0.0898 0.4030
  2 0.0514 0.3370
  3 0.0990 0.4988

```

## Resultado do segundo experimento

Figura 5 – Output das configurações para o segundo experimento

```

# 2021-09-15 13:53:20 -03: Iteration 2 of 3
# experimentsUsedSoFar: 1662
# remainingBudget: 3338
# currentBudget: 1669
# nbConfigurations: 241

```

Figura 6 – Instâncias treinadas

	Instance	Alive	Best	Mean best	Exp so far	W time	rho	KenW	Qvar
x	7	241	1	-39.17200000	241	00:31:55	NA	NA	NA
x	4	241	1	-43.17300000	479	00:31:33	+1.00	1.00	0.0000
x	6	241	1	-48.57300000	717	00:31:39	+1.00	1.00	0.0000
x	2	241	1	-46.59975000	955	00:31:34	+1.00	1.00	0.0000
=	3	241	1	-46.09880000	1193	00:31:32	+1.00	1.00	0.0000
=	5	241	1	-48.64616667	1431	00:32:23	+1.00	1.00	0.0000
=	1	241	1	-48.34942857	1669	00:32:05	+1.00	1.00	0.0000

Figura 7 – Resultados obtidos

```
Best-so-far configuration:          1    mean value:    -48.34942857
Description of the best-so-far configuration:
.ID.  tempo      T .PARENT.
1     1 0.0898 0.403      NA

# 2021-09-15 17:36:07 -03: Elite configurations
# (first number is the configuration ID;
# listed from best to worst according to the sum of ranks):
      tempo      T
1 0.0898 0.4030
2 0.0514 0.3370
3 0.0990 0.4988
```

#### Resultado do terceiro experimento

Figura 8 – Output das configurações

```
# 2021-09-15 17:36:07 -03: Iteration 3 of 3
# experimentsUsedSoFar: 3331
# remainingBudget: 1669
# currentBudget: 1669
# nbConfigurations: 211
```

Figura 9 – Instâncias Treinadas

	Instance	Alive	Best	Mean best	Exp so far	W time	rho	KenW	Qvar
x	8	211	1	-41.49700000	211	00:28:32	NA	NA	NA
x	4	211	1	-44.33550000	419	00:27:46	+1.00	1.00	0.0000
x	6	211	1	-49.34800000	627	00:28:27	+1.00	1.00	0.0000
x	5	211	1	-52.35675000	835	00:28:42	+1.00	1.00	0.0000
=	1	211	1	-51.19920000	1043	00:27:40	+1.00	1.00	0.0000
=	7	211	1	-49.19466667	1251	00:27:46	+1.00	1.00	0.0000
=	2	211	1	-47.97828571	1459	00:27:45	+1.00	1.00	0.0000
=	3	211	1	-47.49287500	1667	00:28:30	+1.00	1.00	0.0000

Figura 10 – Resultados obtidos

```
Best-so-far configuration:          1    mean value:    -47.49287500
Description of the best-so-far configuration:
.ID.  tempo      T .PARENT.
1     1 0.0898 0.403      NA

# 2021-09-15 21:21:22 -03: Elite configurations
# (first number is the configuration ID;
# listed from best to worst according to the sum of ranks):
      tempo      T
1 0.0898 0.4030
2 0.0514 0.3370
3 0.0990 0.4988
```

## Conclusão

Figura 11 -Output final dos melhores parâmetros

```
# Best configurations as commandlines
#(first number is the configuration ID; same order as above):
1  --tempo 0.0898 --T 0.403
2  --tempo 0.0514 --T 0.337
3  --tempo 0.099 --T 0.4988
```

A Figura 11 apresenta os três melhores parâmetros obtidos para o código da busca tabu.