Uma introdução à ferramenta de calibração de parâmetros irace

Débora N. Diniz, Maycon Amaro

O problema

- Muitos algoritmos de otimização são configuráveis, isto é, possuem um grande número de parâmetros que devem ser definidos pelo usuário

O problema

- Muitos algoritmos de otimização são configuráveis, isto é, possuem um grande número de parâmetros que devem ser definidos pelo usuário

- Esta escolha está diretamente relacionada ao seu desempenho

A calibração manual de parâmetros

- Exige muito esforço humano

A calibração manual de parâmetros

- Exige muito esforço humano
- Pode ser tendenciosa e difícil de reproduzir, uma vez que é guiada pela experiência pessoal

A calibração manual de parâmetros

- Exige muito esforço humano
- Pode ser tendenciosa e difícil de reproduzir, uma vez que é guiada pela experiência pessoal
- Poucas configurações são exploradas

A calibração manual de parâmetros

- Exige muito esforço humano
- Pode ser tendenciosa e difícil de reproduzir, uma vez que é guiada pela experiência pessoal
- Poucas configurações são exploradas
- Instâncias usadas tanto para treino, quanto para teste levam a uma avaliação tendenciosa do desempenho

O irace

- Framework para configuração automática de parâmetros que determina a melhor combinação de parâmetros de um algoritmo

O irace

- Framework para configuração automática de parâmetros que determina a melhor combinação de parâmetros de um algoritmo

- Implementado em R

O irace

- Framework para configuração automática de parâmetros que determina a melhor combinação de parâmetros de um algoritmo
- Implementado em R
- Extensão da corrida iterativa Iterated F-Race (I/F-Race)

O irace

- Framework para configuração automática de parâmetros que determina a melhor combinação de parâmetros de um algoritmo
- Implementado em R
- Extensão da corrida iterativa Iterated F-Race (I/F-Race)
- Utiliza o teste estatístico de Friedman

Definição do problema

- Dado um algoritmo parametrizado com *n* parâmetros, tem-se o vetor X_d de parâmetros, com *d = 1, ..., n*, sendo que cada um deles pode assumir diferentes valores

Definição do problema

- Dado um algoritmo parametrizado com *n* parâmetros, tem-se o vetor X_d de parâmetros, com *d = 1, ..., n*, sendo que cada um deles pode assumir diferentes valores

- Uma configuração do algoritmo $\theta = x_1, ..., x_n$ é uma atribuição única de valores dos parâmetros e Θ denota o conjunto possivelmente infinito de todas as configurações do algoritmo

Definição do problema

- Dado um algoritmo parametrizado com *n* parâmetros, tem-se o vetor X_d de parâmetros, com *d = 1, ..., n*, sendo que cada um deles pode assumir diferentes valores

- Uma configuração do algoritmo $\theta = x_1, ..., x_n$ é uma atribuição única de valores dos parâmetros e Θ denota o conjunto possivelmente infinito de todas as configurações do algoritmo

- Requer um conjunto de instâncias com uma probabilidade associada pelo qual $I = i_1, i_2, ...$ é selecionado aleatoriamente

Definição do problema

- Requer um custo *c(θ, i)*, que associa um valor a cada configuração aplicada em uma dada instância

Definição do problema

- Requer um custo *c(θ, i)*, que associa um valor a cada configuração aplicada em uma dada instância

- O critério a ser otimizado é uma função $F(\theta) : \Theta \rightarrow \mathbb{R}$ de custo da configuração θ em relação à distribuição da variável aleatória I.

Definição do problema

- Requer um custo *c(θ, i)*, que associa um valor a cada configuração aplicada em uma dada instância
- O critério a ser otimizado é uma função $F(\theta) : \Theta \rightarrow \mathbb{R}$ de custo da configuração θ em relação à distribuição da variável aleatória I.
- O objetivo da configuração automática é encontrar a melhor configuração θ* que minimiza F(θ)

Passos do método

1. Gerar uma amostra de novas configurações de parâmetros de acordo com uma distribuição específica

2. Selecionar as melhores configurações da amostra atual por meio de corrida

3. Atualizar a distribuição da amostragem de forma que restrinja a amostragem ao redor dos valores das melhores configurações

4. Repete os anteriores até um critério de parada

Algorithm Algorithm outline of iterated racing.

Require: $I = \{I_1, I_2, \ldots\} \sim \mathcal{I}$,
parameter space: X,
cost measure: $C(\theta, i) \in \mathbb{R}$,
tuning budget: B

1: $\Theta_1 = \text{SampleUniform}(X)$
2: $\Theta^{\text{elite}} = \text{Race}(\Theta_1, B_1)$
3: $j = 1$
4: **while** $B^{\text{used}} \leq B$ **do**
5: $\quad j = j + 1$
6: $\quad \Theta^{\text{new}} = \text{Sample}(X, \Theta^{\text{elite}})$
7: $\quad \Theta_j = \Theta^{\text{new}} \cup \Theta^{\text{elite}}$
8: $\quad \Theta^{\text{elite}} = \text{Race}(\Theta_j, B_j)$
9: **end while**
10: **Output:** Θ^{elite}

Iterated Racing

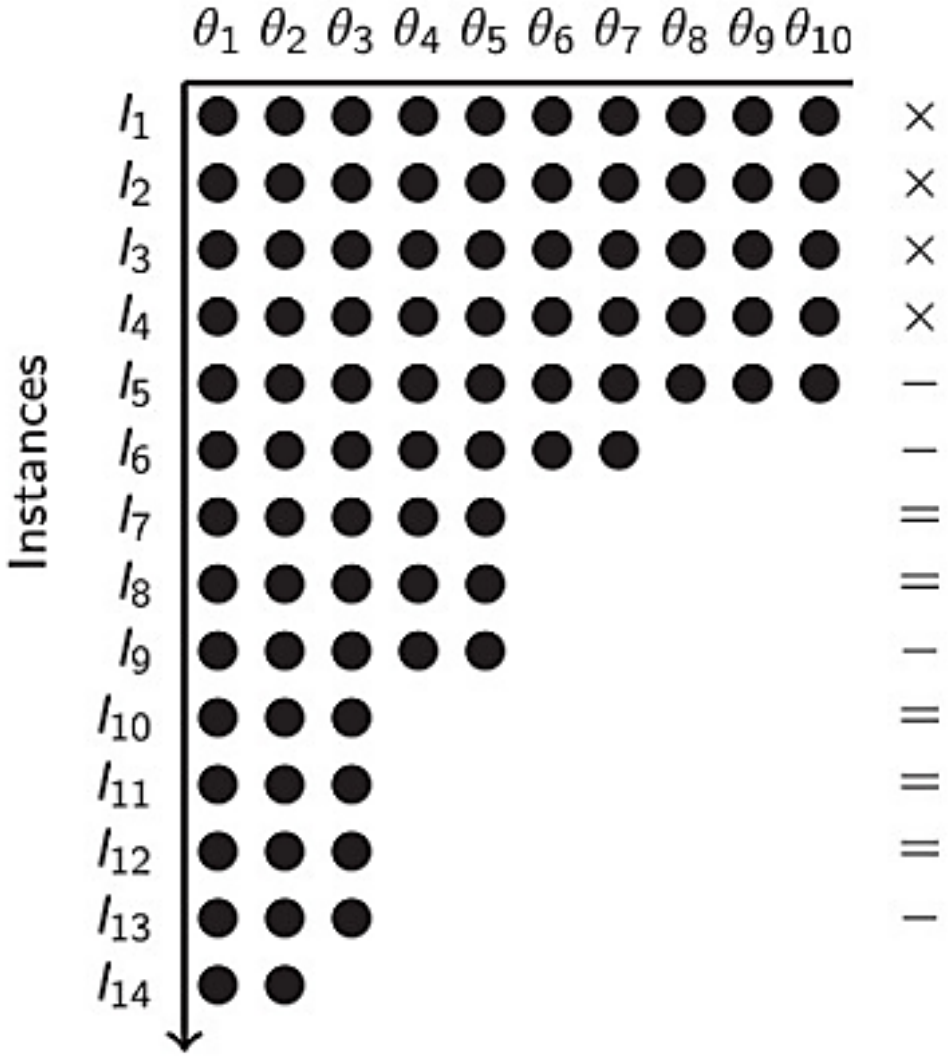
- Cada ponto é a avaliação de uma configuração aplicada a uma instância

  x significa que nenhum teste estatístico é feito

  - significa que o teste descartou pelo menos uma configuração

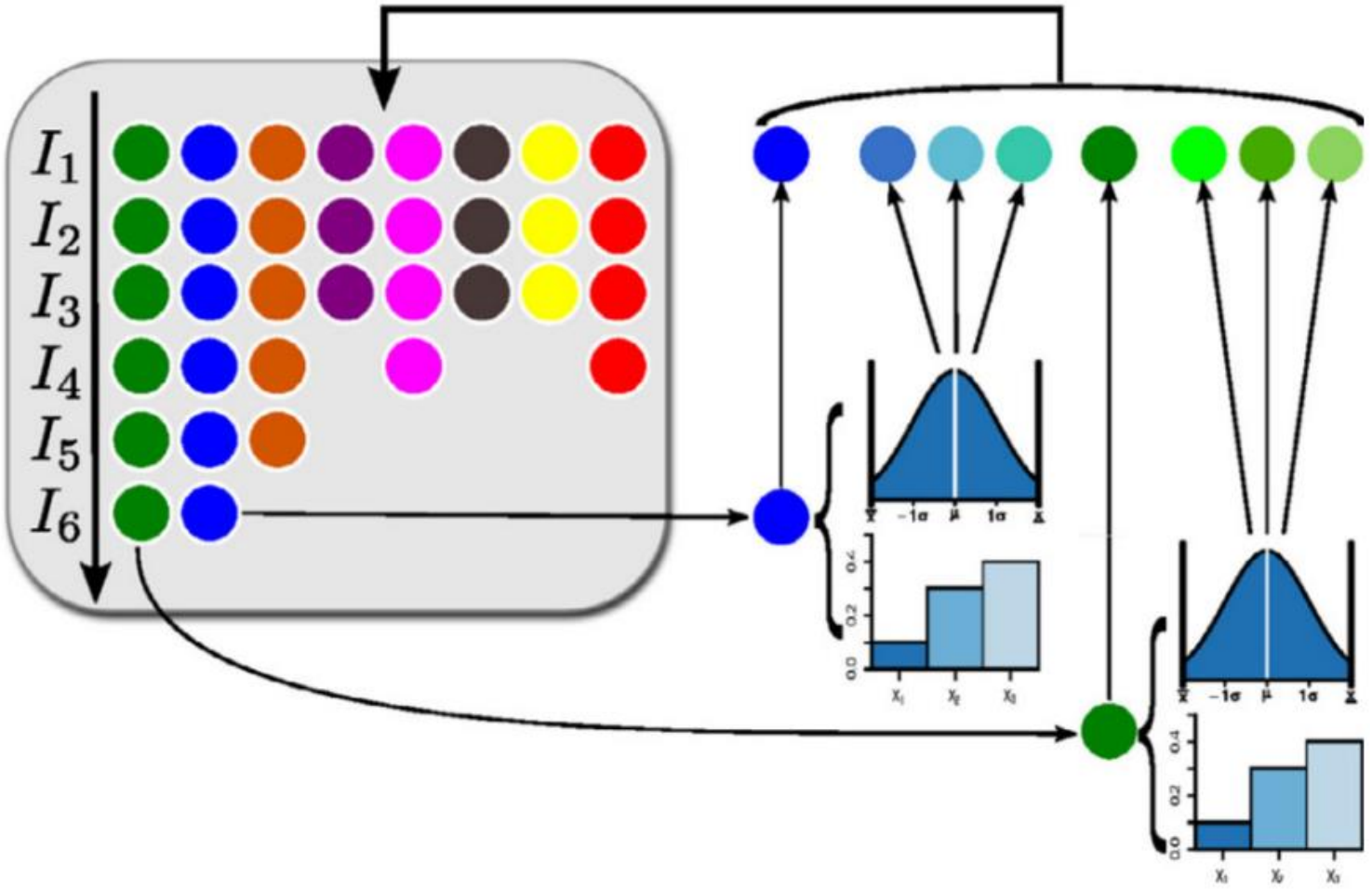
  = significa que o teste não descartou nenhuma configuração

- $T^{first} = 5$
- $T^{each} = 1$



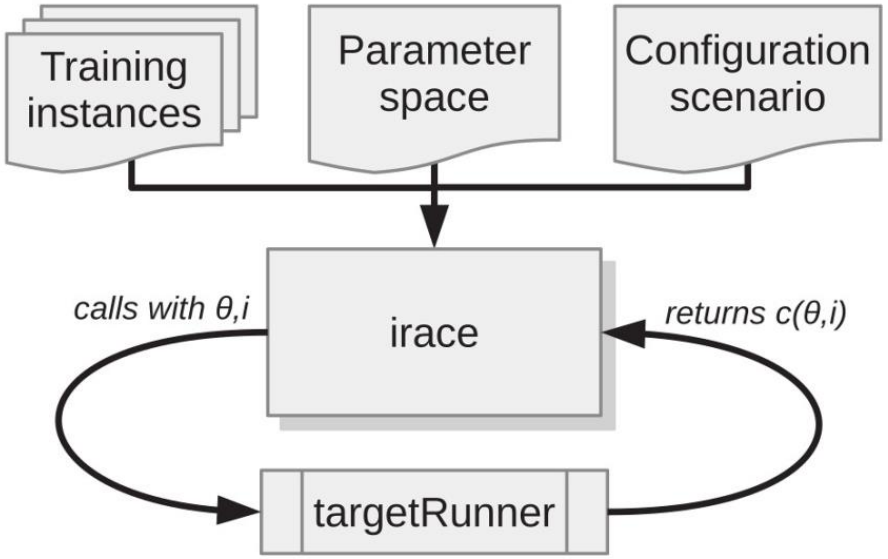
Iterated Racing

- Atualização da amostragem



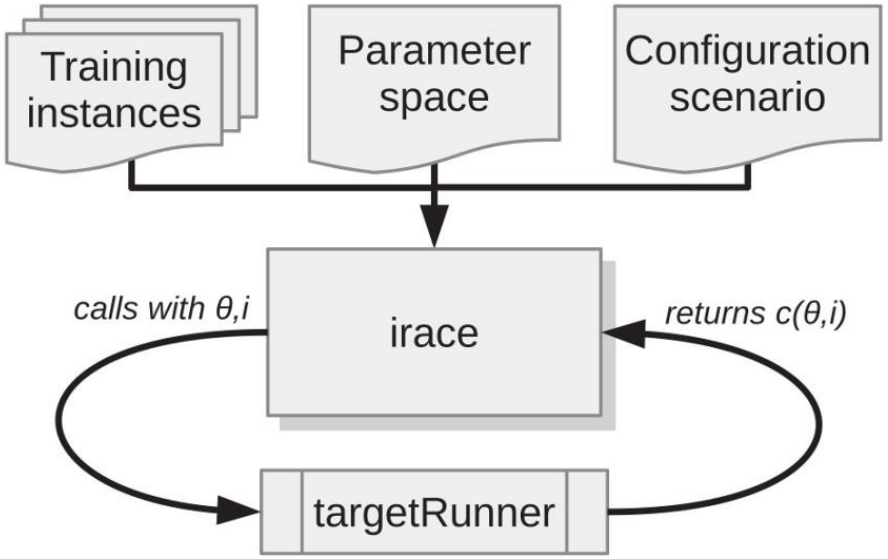
O irace

- Conjunto de instâncias:
  - Conjunto finito usado para representar o problema tratado

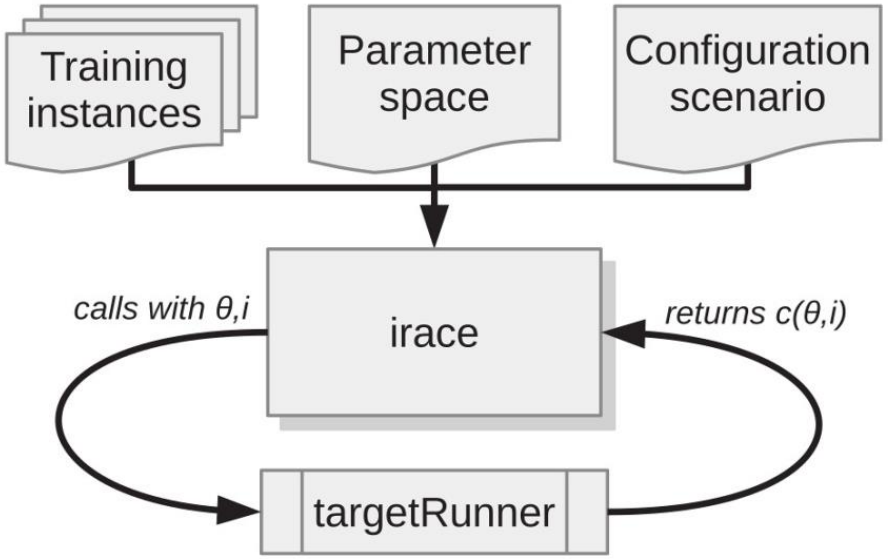


O irace

- Conjunto de instâncias:
  - Conjunto finito usado para representar o problema tratado
  - É configurado no *scenario.txt*



O irace

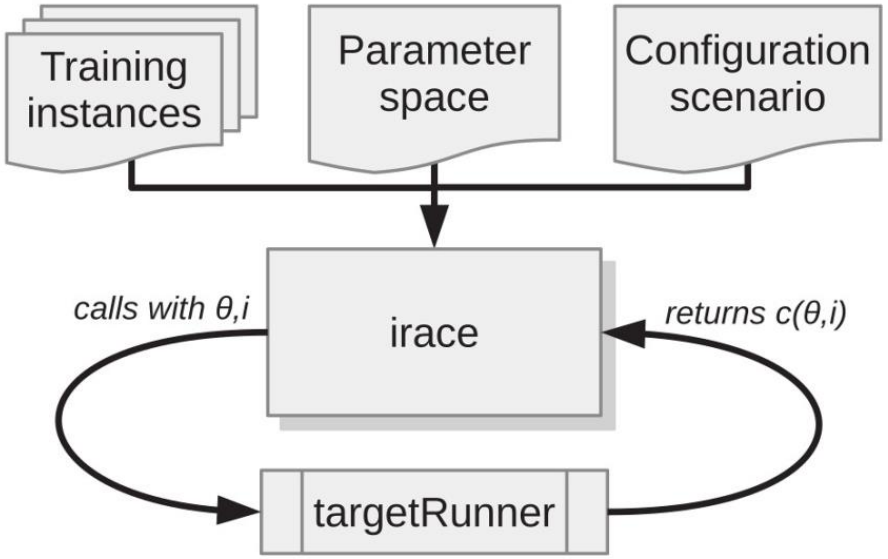


- Conjunto de instâncias:
  - Conjunto finito usado para representar o problema tratado

  - É configurado no *scenario.txt*
    - Como diretório das instâncias em *trainInstancesDir*

OU

O irace

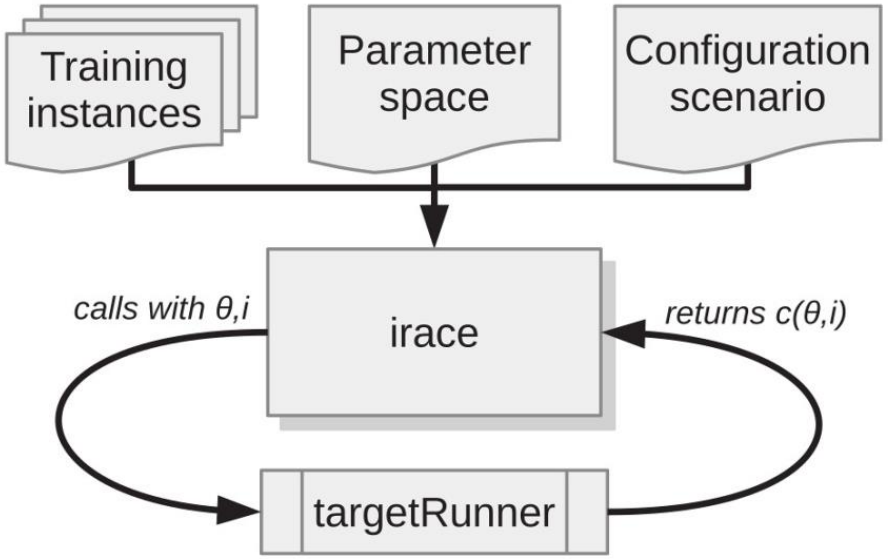


- Conjunto de instâncias:
  - Conjunto finito usado para representar o problema tratado
  - É configurado no *scenario.txt*
    - Como diretório das instâncias em *trainInstancesDir*

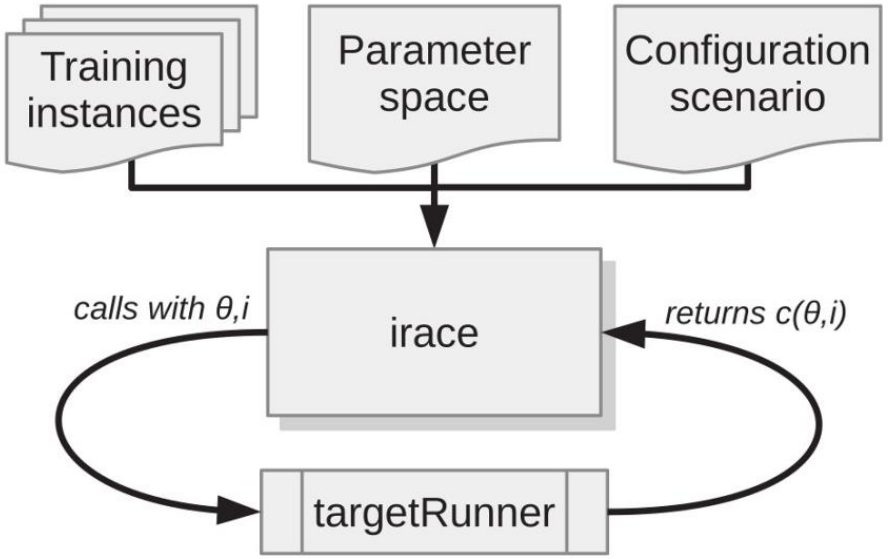
      OU
    - Como um arquivo de instâncias em *trainInstancesFile*

O irace

- Espaço de parâmetros:
  - Possíveis parâmetros que podem ser configurados, seus tipos, valores e constantes



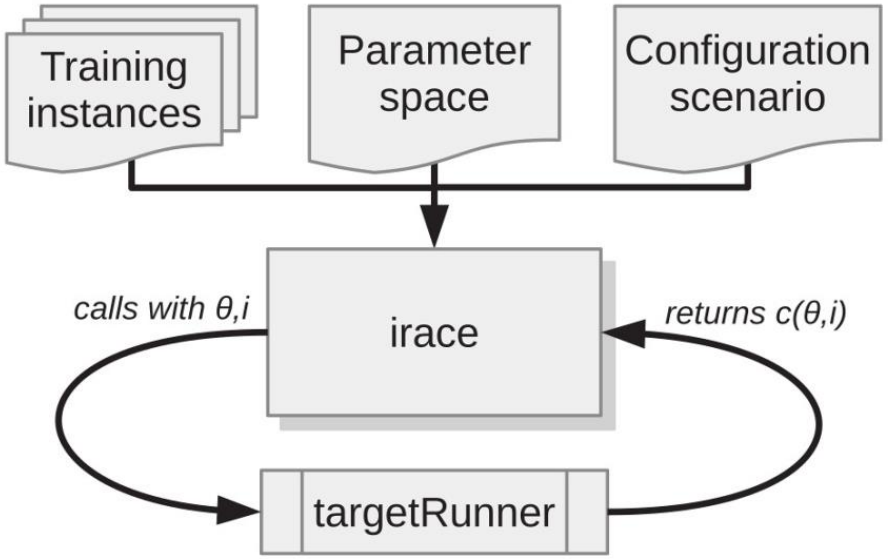
O irace



- Espaço de parâmetros:
  - Possíveis parâmetros que podem ser configurados, seus tipos, valores e constantes
  - Tipos: categóricos (**c**), inteiros (**i**), ordinais (**o**), reais (**r**)

```
# name          switch              type    values          [conditions (using R syntax)]

localsearch     "--localsearch "    c       (0, 1, 2, 3)
alpha           "--alpha "          r       (0.00, 5.00)
beta            "--beta "           r       (0.00, 10.00)
ants            "--ants "           i       (5, 100)
```

O irace

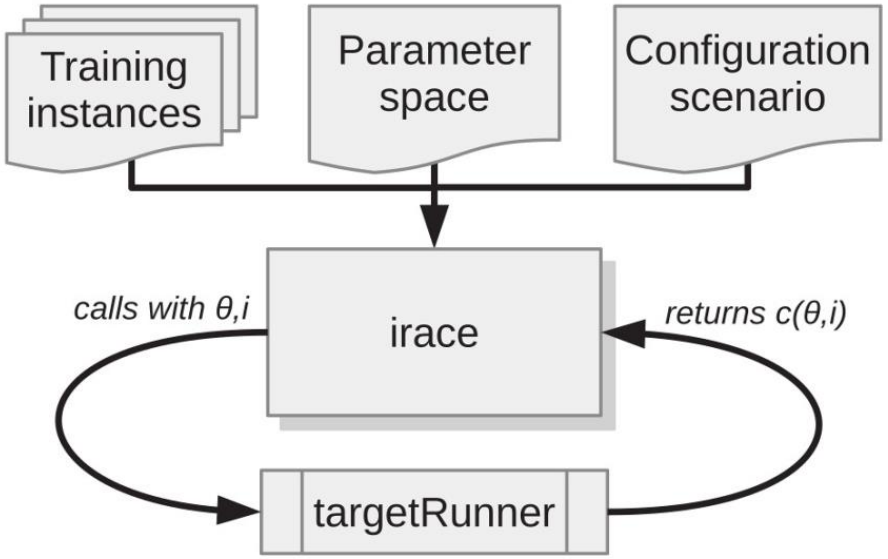


- Espaço de parâmetros:
  - Possíveis parâmetros que podem ser configurados, seus tipos, valores e constantes
  - Tipos: categóricos (**c**), inteiros (**i**), ordinais (**o**), reais (**r**)

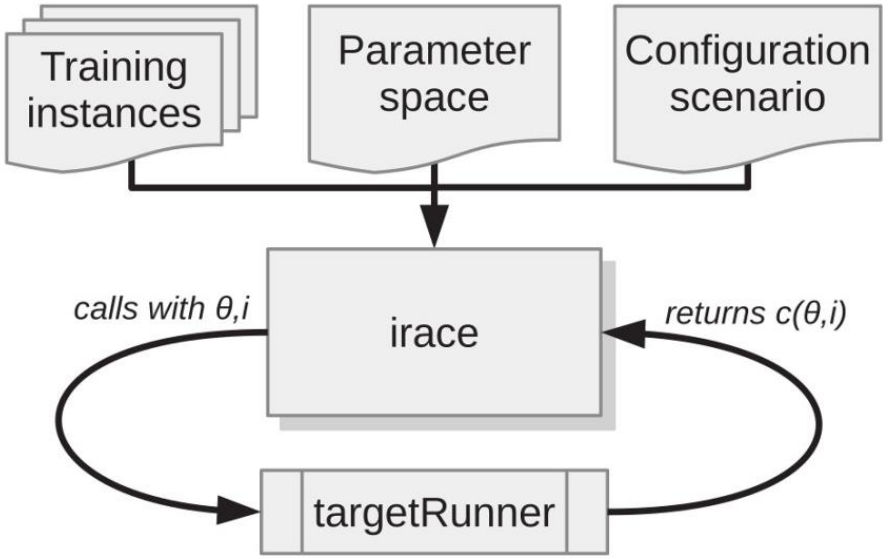
```
# name          switch              type      values          [conditions (using R syntax)]

localsearch     "--localsearch "    c         (0, 1, 2, 3)
alpha           "--alpha "          r         (0.00, 5.00)
beta            "--beta "           r         (0.00, 10.00)
ants            "--ants "           i         (5, 100)
```

O irace

- Cenário:
  - Configurações como arquivo de definição de parâmetros, budget de execução e número de candidatos



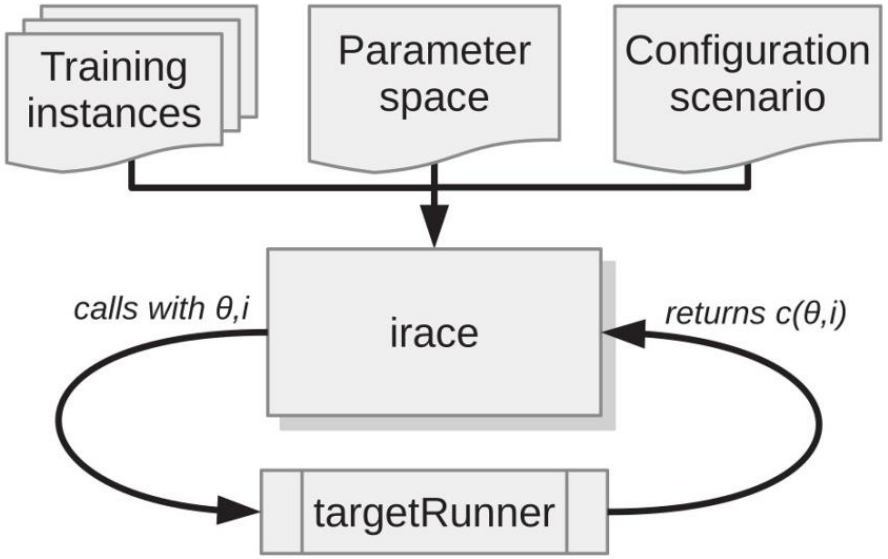
O irace



- Cenário:
  - Configurações como arquivo de definição de parâmetros, budget de execução e número de candidatos

- ***Digits*:** número de casas decimais a serem consideradas (default 4)

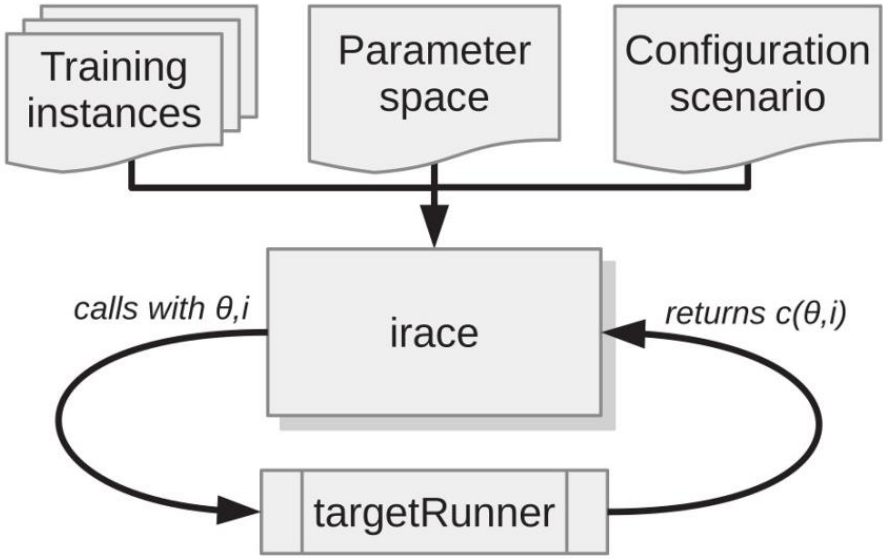
O irace



- Cenário:
  - Configurações como arquivo de definição de parâmetros, budget de execução e número de candidatos

- ***Digits*****:** número de casas decimais a serem consideradas (default 4)
- ***MaxExperiments*****:** número máximo de execuções do algoritmo

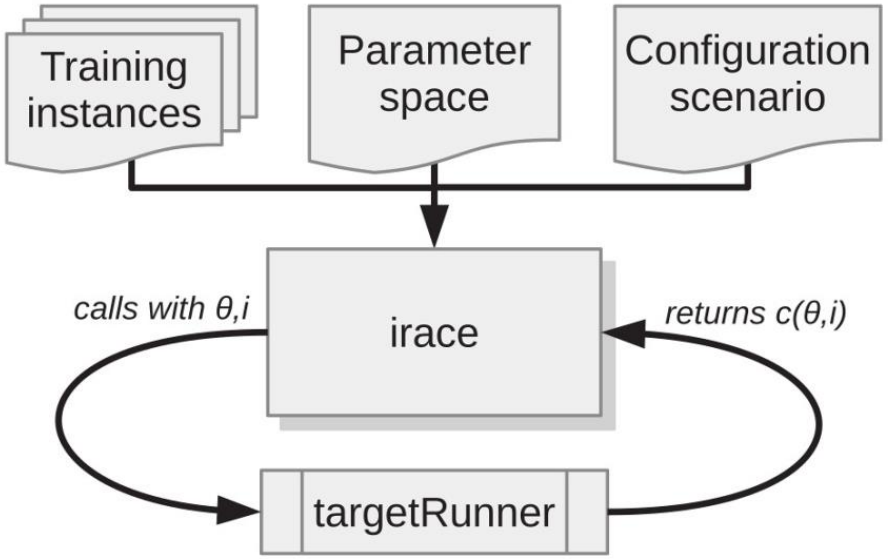
O irace



- Cenário:
  - Configurações como arquivo de definição de parâmetros, budget de execução e número de candidatos

- ***Digits*****:** número de casas decimais a serem consideradas (default 4)
- ***MaxExperiments*****:** número máximo de execuções do algoritmo
- ***TypeTest*****:** teste estatístico a ser utilizado (F-test ou t-test)

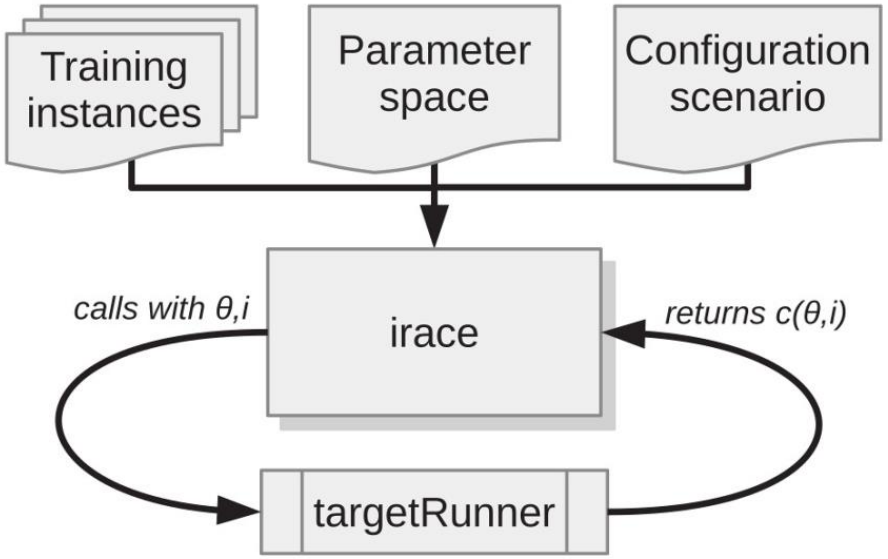
O irace



- Cenário:
  - Configurações como arquivo de definição de parâmetros, budget de execução e número de candidatos

- ***Digits*****:** número de casas decimais a serem consideradas (default 4)
- ***MaxExperiments*****:** número máximo de execuções do algoritmo
- ***TypeTest*****:** teste estatístico a ser utilizado (F-test ou t-test)
- ***FirstTest*****:** especifica a quantidade de instâncias que serão analisadas antes do primeiro teste de eliminação

O irace

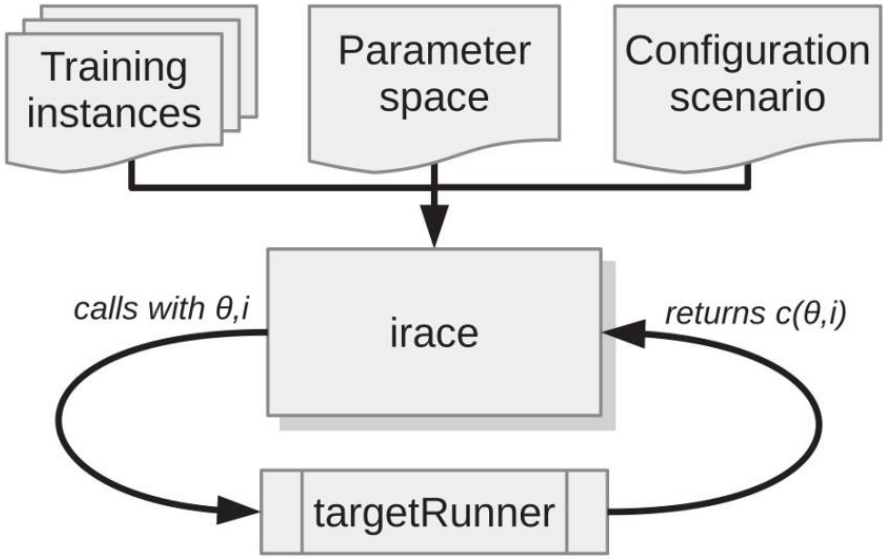


- Cenário:
  - Configurações como arquivo de definição de parâmetros, budget de execução e número de candidatos

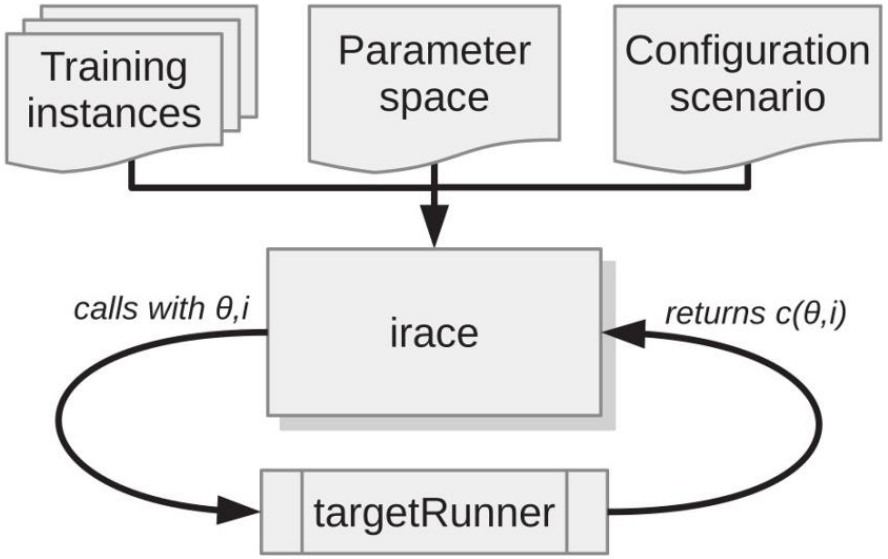
- ***Digits*****:** número de casas decimais a serem consideradas (default 4)
- ***MaxExperiments*****:** número máximo de execuções do algoritmo
- ***TypeTest*****:** teste estatístico a ser utilizado (F-test ou t-test)
- ***FirstTest*****:** especifica a quantidade de instâncias que serão analisadas antes do primeiro teste de eliminação
- ***EachTest*****:** especifica a quantidade de instâncias que serão analisadas entre teste de eliminação

O irace

- TargetRunner:
  - Script que chama o algoritmo a ser executado para calibração



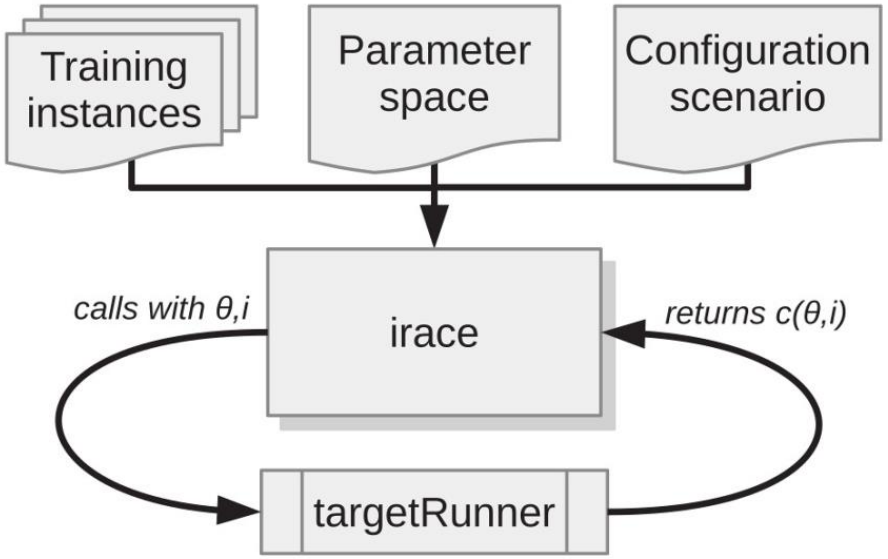
O irace



- TargetRunner:
  - Script que chama o algoritmo a ser executado para calibração

- O caminho do executável deve ser colocado em *EXE* no arquivo target-runner

O irace

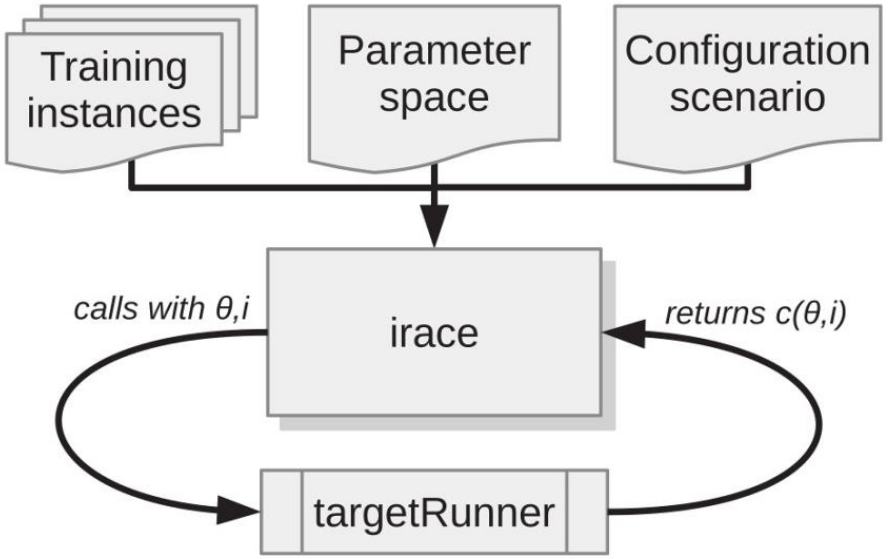


- TargetRunner:
  - Script que chama o algoritmo a ser executado para calibração

- O caminho do executável deve ser colocado em *EXE* no arquivo target-runner

- A resposta do algoritmo a ser calibrado deverá ser apenas um valor custo

O irace



- TargetRunner:
  - Script que chama o algoritmo a ser executado para calibração

- O caminho do executável deve ser colocado em *EXE* no arquivo target-runner

- A resposta do algoritmo a ser calibrado deverá ser apenas um valor custo

- O irace calibra problemas de minimização. Para maximização, multiplicar o custo por -1.

Instalação

- Fazer a instalação do R:
  https://www.r-project.org/ *versões superiores à 2.15.0

- Fazer a instalação do irace e configurar as variáveis de ambiente:
  http://iridia.ulb.ac.be/irace/README.html

- Baixar o exemplo que será utilizado:
  https://mayconamaro.github.io/irace/

Exemplo prático - GRASP

procedimento $GRASP(f(.), g(.), N(.), GRASPmax, s)$
1 $f^{\star} \leftarrow \infty$;
2 para $(Iter = 1, 2, \ldots, GRASPmax)$ faça
3 $Construcao(g(.), \alpha, s)$;
4 $BuscaLocal(f(.), N(.), s)$;
5 se $(f(s) < f^{\star})$ então
6 $s^{\star} \leftarrow s$;
7 $f^{\star} \leftarrow f(s)$;
8 fim-se;
9 fim-para;
10 $s \leftarrow s^{\star}$;
11 Retorne s;
fim $GRASP$

Uma introdução à ferramenta de calibração de parâmetros irace

Débora N. Diniz, Maycon Amaro