Tabela 1: Resultados algoritmos EDA.

Alg.	Inst.	Opt	$\overline{m}$	%	t	$\pm\sigma_t$	i	$\pm\sigma_i$	It.	It/R	t/R
B&D	flat100-1	1117	1101.1	98.6	43.8	74.4	313.9	720.5	1584.3	883.3	207.2
B&D	flat50-1	545	539.3	98.9	117.9	95.8	1371.5	1089.3	3509.6	714.2	61.8
B&D	flat75-1	840	828.4	98.6	80.5	83.1	633.0	952.2	1890.2	712.4	128.7
B&D	par8-5-c	298	295.6	99.2	133.1	84.9	5778.1	3739.5	12708.2	839.4	20.4
MIMIC	flat100-1	1117	1100.3	98.5	52.5	79.4	356.8	766.6	1288.3	830.9	246.6
MIMIC	flat50-1	545	539.5	99.0	108.7	85.1	1097.1	845.7	3129.1	710.1	68.5
MIMIC	flat75-1	840	827.5	98.5	91.9	101.1	584.5	785.5	2144.4	866.9	160.3
MIMIC	par8-5-c	298	295.9	99.3	119.8	97.2	5084.2	4378.6	12316.2	679.6	16.7
UMDA	flat100-1	1117	1104.3	98.9	83.5	64.1	7052.6	8377.1	22125.9	1419.6	22.7
UMDA	flat50-1	545	540.5	99.2	102.5	96.6	13408.8	13590.0	39757.9	1055.0	8.9
UMDA	flat75-1	840	830.0	98.8	136.1	82.0	10684.5	6870.8	26631.8	1334.9	16.7
UMDA	par8-5-c	298	296.7	99.6	135.3	83.8	31958.0	20001.8	75546.3	1184.4	5.2

Soluções 3

Exercício 1 (Algoritmos de estimação de distribuição)
A Tabela 1 mostra os tempos e iterações médios e os desvios dos três algoritmos. O tempo limite de todas execuções foi 300s. Para aproveitar melhor o tempo, foram adicionados reinícios: ao detectar a convergência do modelo probabilístico (i.e. uma probabilidade alta de produzir sempre a mesma solução) o modelo é resetado para um modelo uniforme. Em experimentos simples preliminares o tamanho da população foi fixada para 1000. (Em uma aplicação real este valor deveria ser calibrado). Seguindo a definição do UMDA, os melhores 50% da população foram consideradas soluções "boas". Para ficar comparável estes valores foram mantidos para todos três algoritmos.
Nenhum algoritmo encontrou uma solução factível, e o tempo limite de 300s sempre foi alcançado. Por isso tempo e iterações na tabela ($t \pm \sigma_t$, $i \pm \sigma_i$) são até encontrar a melhor solução. Adicionalmente, a tabela mostra o número de claúsulas (Opt), o número e percentagem média de claúsulas satisfeitas ($\overline{m}$ e %), o número médio de iterações globais (It., uma iteração inclui a amostragem e reconstrução do modelo), e iterações e tempo médio entre reinícios (It/R, t/R).
Podemos ver que nenhum algoritmo encontrou uma solução factível, mas todos ficaram muito perto, sempre com mais que 98.5% das claúsulas satisfeitas. Então em geral os modelos probabilísticos não são suficientes para encontrar soluções factíveis. Comparando os algoritmos, o UMDA domina MIMIC e Baluja e Davies (B&D) mas entre os dois últimos não tem um algoritmo claramente melhor.
Os resultados podem ser explicados observando as demais métricas: os algoritmos mais sofisticados precisam mais tempo para convergir (maior t/R) mas conseguem pelo menos uma ordem de grandeza menos iterações que o UMDA (menor It.). Isso sugere que o desempenho melhor do UMDA é explicado pelo grande número de amostras e os modelos melhores não conseguem compensar isso. (Um teste para verificar essa hipótese fixar

Tabela 2: Configurações testadas.

Configuração	Duração tabu	Solução inicial	Compilação
1	$0.1n$	Aleatória	-O
2	$0.1n$	Aleatória	-O3
3	$0.2n$	Aleatória	M_1
4	M_2	Gulosa	M_1

o número de iterações; além disso um teste com somente amostragem aleatória seria indicado para avaliar a contribuição dos modelos.)

Exercício 2 (Testes estatísticos)
O plano de teste tem que incluir diversas instâncias. Como o método GSAT/Tabu é randomizada (por selecionar um dos melhores vizinhos aleatoriamente) vamos executar todas configurações múltiplas vezes e comparar os tempos médios. Isso permitiria também usar uma busca tabu com duração tabu aleatória, mas para reduzir o número de testes vamos usar uma duração tabu de $0.1n$ e $0.2n$. Concretamente o plano de teste é

Instâncias Classes de instâncias uf75, flat75, RTI_k3_n100_m429, e instâncias 20, 40, 60, 80, 100 de cada classe.

Replicações 30 com sementes $1, \ldots, 30$ e tempo limite 120s por replicação.

Configurações Testaremos as configurações da tabela 2, com M_1 o melhor entre -O e -O3 conforme o resultado do primeiro teste e M_2 o melhor entre $0.1n$ e $0.2n$ conforme o resultado do segundo teste.

A Tabela 3 mostra os resultados de todos testes.

Teste 1 e 2 O primeiro teste compara as colunas 1 e 2 da Tabela 3. A otimização completa é (um pouco) mais rápida em 10 dos 15 testes, que não é suficiente para rejeitar a hipótese nula que eles precisam o mesmo tempo usando um teste de sinal ($p = 0.15$), sendo a hipótese alternativa que a otimização completa precisa menos tempo. Como esperado o número de passos foi idêntico em todas instâncias resolvidas (882 do 900 testes), logo não podemos concluir rejeitar a hipótese nula que ambas níveis de otimização precisam o mesmo número de passos. Nos vamos manter a otimização completa para os próximos testes.

Teste 3 A Tabela 3 mostra os resultados do terceiro teste (colunas com configurações 2 e 3, com M_1 = -O3). A configuração 2 é mais rápida em 8 dos 15 casos, e novamente não podemos rejeitar a hipótese nula que eles precisam o mesmo tempo (isso também com um teste de postos com sinais), mesmo com médias diferentes. Vamos escolher uma duração tabu $0.2n$ para o último teste.

Tabela 3: Comparação otimização simples (-O) e otimização completa (-O3).

Nome	Inst.	Tempo	Tempo	Tempo	Tempo
Configuração		1	2	3	4
RTI_k3_n100_m429	20	18.93	18.53	18.67	30.37
RTI_k3_n100_m429	40	4104.20	4101.83	13.40	17.70
RTI_k3_n100_m429	60	8.90	8.57	18.07	16.87
RTI_k3_n100_m429	80	18.67	18.27	8.93	7.17
RTI_k3_n100_m429	100	13.50	13.23	15.33	11.27
flat75	20	13.87	13.87	318.17	424.83
flat75	40	129.73	129.73	3394.07	3690.30
flat75	60	10.60	10.63	263.70	314.63
flat75	80	46.47	46.60	1209.10	832.33
flat75	100	17.70	17.90	315.23	487.50
uf75	20	24010.37	24010.33	2.70	2.53
uf75	40	5509.67	5474.10	24.13	14.70
uf75	60	19.40	19.03	3.10	1.73
uf75	80	6.03	5.97	1.13	1.23
uf75	100	4258.80	4253.13	5.10	4.57
Médias		2545.79	2542.78	374.06	390.52

Teste 4 A Tabela 3 mostra os resultados do quarto teste (colunas com configurações 3 e 4, com $M_2 = 0.2n$). A configuração 4 é mais rápida em 8 dos 15 casos, e novamente não podemos rejeitar a hipótese nula que eles precisam o mesmo tempo (isso também com um teste de postos com sinais).

Complexidade empírica A complexidade empírica do configuração 3 no modelo linear é

$$1.22\mu\text{s} \cdot n^{6.63} m^{-2.35}.$$

Para testar a linearidade em m a hipótese nula é $\beta = 1$ (com $\hat{\beta} = -2.35$) e a hipótese alternativa $\beta \neq 1$. Um teste t (de acordo com o exemplo 6.14 das notas) rejeita a hipótese nula num nível de $p < 0.001$, usando

```
2*pt(t,n-2,lower.t=F)
```

para o teste bicaudal.
(O modelo linear a princípio é suspeito, por se tratar de um problema NP-completo. Logo a hipótese exponencial seria mais adequada. Neste caso obtemos o modelo

$$70\text{ms} \cdot 0.96^n 1.02^m.$$

Ambos os modelos não são muito confiáveis por serem baseados em uma amostra pequena demais.)

Exercício 3 (Calibração de parâmetros com corridas)

Metodologia Continuando a lista 2, vamos calibrar os parâmetros $d, D \in [0, 1)$, $d < D$ que definem uma duração tabu mínima $d_{\min}(n) = \lfloor dn \rfloor$ e máxima $d_{\max}(n) = \lfloor Dn \rfloor$. Uma configuração (`parameters.txt`) correspondente é

```
# name switch type values [| conditions]
d "--ptenure    " i (0,100)
D "--pmaxtenure " i (0,100)
```

com configurações proibidas (`forbidden.txt`)

```
d>D
```

A segunda lista testou 10 níveis em 4 instâncias com 5 replicações para um total de 200 testes. O mesmo número foi definido como número máximo de testes para o irace com as mesmas 4 instâncias. O tempo limite de 120s foi mantido. Como o objetivo é minimizar o tempo, foi usado uma versão elitista com "adapative capping". Executamos os testes em paralelo e informamos que o algoritmo não é determinístico. O resumo dos principais parâmetros (`scenario.txt`) então é

```
parallel=8
deterministic=0
elitist = 1
capping = 1
boundMax = 120
```

Outras observações:

- Instâncias de calibração e teste devem ser diferentes; eles são os mesmos somente neste exercício.
- Uma possibilidade para melhor o resultado é informar a configuração $[0.1, 0.4]$ encontrado no primeiro teste. Isso não foi feito para poder comparar melhor os métodos.

Resultados A melhor configuração encontrada depois de um tempo total de 11 minutos de calibração é $[0.02, 0.22]$. A Tabela 4 compara os resultados com a segunda lista, onde "manual" é GSAT/Tabu com a configuração manual, "irace" usando os parâmetros encontrados pelo irace. (Como foi usada uma outra máquina e versão de código o experimento de lista 2 foi repetido.)
Não tem uma clara vantagem para uma das duas configurações: os parâmetros encontrados pelo irace funcionam melhor para instâncias do tipo "flat", mas pior para par8-5-c. Como observado na segunda lista, os dois tipos de instâncias tem diferentes domínios de parâmetros ótimos.

Tabela 4: Resultados calibração GSAT/Tabu.

Cal.	Inst.	t	$\pm\sigma_t$	i	$\pm\sigma_i$
irace	flat100-1	756.8	467.3	40.4	24.3
irace	flat50-1	23.8	21.4	2.6	2.4
irace	flat75-1	140.2	83.8	7.7	4.3
irace	par8-5-c	1830.6	3070.4	739.9	1392.8
manual	flat100-1	4016.8	2933.1	335.0	285.8
manual	flat50-1	52.0	57.9	5.2	5.6
manual	flat75-1	1218.2	1384.9	111.6	157.8
manual	par8-5-c	353.4	403.0	72.0	80.4

Para aplicar o teste estatístico, não podemos supor a normalidade dos dados, logo um teste não-paramêtrico é adequado. (Uma alternativa seria verificar a normalidade é usar uma ANOVA.) As instâncias formam os blocos do teste, e os diferentes algoritmos podem ser vistos como tratamentos. Um teste adequado neste caso é um teste de Friedman. Porém, o teste de Friedman é definido somente para uma replicação por célula experimental. Um teste que permite mais replicações é o teste de Mack-Skillings (Hollander et al. 2014, cáp. 7.9) (não visto em aula).
O teste produz um valor $p = 0.0246$, ou seja a diferença é não significativamente diferente para um nível de confiância de $\alpha = 0.01$.

Referências

Hollander, M., D. A. Wolfe e E. Chicken (2014). *Nonparametric statistical methods*. 3ª ed. Wiley.