



Modelagem Computacional no
Sertão Mineiro



PPGMCS
Programa de Pós-Graduação em
Modelagem Computacional e Sistemas

UM ALGORITMO GENÉTICO–GRASP HÍBRIDO PARA O BALANCEAMENTO DE CARGA NO ESCALONAMENTO DE MOTORISTAS: ANÁLISE DE CUSTO E QUALIDADE

Luis Otavio Machado Soares¹ - lotmsoares@gmail.com

¹Universidade Estadual de Montes Claros, UNIMONTES - Montes Claros, MG, Brasil

Abstract. *Este trabalho investiga o efeito de incorporar a fase de busca local do GRASP (Greedy Randomized Adaptive Search Procedure) aos descendentes de cada geração de um Algoritmo Genético (AG) aplicado a um problema de escalonamento de motoristas, formulado como balanceamento de carga com limites brando e rígido de jornada. A busca local é feita nos melhores descendentes de cada geração e comparada com os motoristas com uma maior carga. Mostramos analiticamente que (i) a fase GRASP acrescenta custo de tempo por geração, linear no número s de descendentes; e (ii) o operador de busca local é monótono, de modo que, sob seleção elitista, a melhor aptidão nunca piora. Experimentos em 30 execuções independentes confirmam a análise: o híbrido reduz o espalhamento de carga em cerca de 49%, ao custo de um fator de retardo de aproximadamente $33\times$ no tempo de computação.*

Keywords: GRASP, Algoritmo genético, Busca local, Escalonamento de motoristas

1. INTRODUÇÃO

A alocação de viagens (*routes*) a motoristas de modo a distribuir de forma equilibrada a jornada diária é uma tarefa recorrente no planejamento de sistemas de transporte. Na prática, é um problema com diversas condições e limitações quanto a horários e locais de início e fim de cada rota, mas, como neste estudo, quando as viagens são independentes e todos os motoristas são equivalentes, o problema se reduz a distribuir um conjunto de tarefas de duração conhecida entre máquinas idênticas de modo a equilibrar as cargas, uma variante do problema de escalonamento em máquinas paralelas idênticas, que é NP-difícil (Garey & Johnson, 1979). Não se conhece, portanto, algoritmo exato de tempo polinomial, o que motiva o uso de metaheurísticas.

Uma destas metaheurísticas é a do Algoritmo Genético (AG), que é uma das metaheurísticas mais difundidas para problemas combinatórios (Holland, 1975; Goldberg, 1989). Ao mesmo tempo, o *Greedy Randomized Adaptive Search Procedure* (GRASP) (Feo & Resende, 1989; Feo & Resende, 1995) é um procedimento que combina uma construção gulosa e aleatorizada

com uma *fase de busca local* que refine a solução até atingir um ótimo local nos vizinhos de uma solução encontrada.

Neste artigo aplicamos, de forma direta, a fase de busca local do GRASP aos descendentes gerados a cada geração de um AG para o escalonamento de motoristas. A contribuição é analisada menos no algoritmo em si, que é uma hibridização clássica de um AG, e mais na sua *caracterização analítica*: (a) mostramos que a fase GRASP acrescenta um custo de tempo por geração, linear no número de descendentes s ; (b) mostramos que a busca local é monótona e, sob seleção elitista, não piora a melhor aptidão; e (c) validamos empiricamente que o ganho de qualidade é estatisticamente significativo, ainda que acompanhado de um aumento considerável no tempo de execução.

2. METODOLOGIA

2.1 FORMULAÇÃO DO PROBLEMA

Sejam N viagens, cada uma com duração $d_i > 0$ ($i = 1, \dots, N$), e D motoristas. Uma escala é um vetor de atribuição $x \in \{1, \dots, D\}^N$, em que $x_i = j$ indica que a viagem i foi alocada ao motorista j . A carga do motorista j é

$$L_j(x) = \sum_{i: x_i=j} d_i. \quad (1)$$

Denotando por $\tau_b = 480$ min o *weak limit*, ou limite brando, de jornada e por $\tau_r = 500$ min o *hard limit*, ou limite rígido, define-se a função objetivo com penalidade

$$f(x) = \underbrace{\left(\max_j L_j - \min_j L_j \right)}_{\text{espalhamento}} + 5 \sum_{j=1}^D \max(L_j - \tau_b, 0) + 100 \sum_{j=1}^D \max(L_j - \tau_r, 0). \quad (2)$$

O primeiro termo mede o desbalanceamento (*espalhamento*) entre motoristas; o segundo penaliza suavemente jornadas acima de τ_b ; e o terceiro penaliza fortemente violações do limite rígido. A violação rígida

$$g(x) = \sum_{j=1}^D \max(L_j - \tau_r, 0) \quad (3)$$

deve ser nula para que a escala seja viável.

2.2 COMBINANDO O AG COM O GRASP

O AG básico opera sobre populações de μ escalas codificadas diretamente como vetores $x \in \{1, \dots, D\}^N$. Seus componentes são:

Inicialização. A população inicial parte de uma solução gulosa LPT (*Longest Processing Time*): as viagens são ordenadas por duração decrescente e cada uma é atribuída ao motorista de menor carga corrente. A LPT fornece um ponto de partida já razoável ao organizar as escalas iniciais; as demais escalas são cópias perturbadas dessa solução.

Operadores genéticos. Recombinação uniforme entre pares de pais e mutação por reatribuição gene-a-gene com baixa probabilidade, produzindo s descendentes por geração. A seleção é elitista do tipo $(\mu + s)$: pais e descendentes são combinados e ordenados lexicograficamente por viabilidade $g(x)$ e, em seguida, por aptidão $f(x)$, preservando-se os μ melhores.

Fase GRASP (busca local por transferência). A cada geração, aplica-se a cada descendente uma busca local sobre a *vizinhança de transferência*. Seja $h = \arg \max_j L_j$ o motorista mais carregado; um vizinho de x é obtido transferindo-se uma viagem r de h para outro motorista $t \neq h$. Formalmente, $\mathcal{N}(x) = \{x^{r \rightarrow t} : x_r = h, t \neq h\}$, onde $x^{r \rightarrow t}$ coincide com x exceto por $x_r^{r \rightarrow t} = t$. A busca aceita a primeira transferência que reduza f em mais de δ e reinicia a varredura quando nenhuma transferência a partir do motorista mais carregado melhora f , ou após κ passos. O impacto de cada movimento é avaliado incrementalmente: transferir r de h para t atualiza apenas $L_h \leftarrow L_h - d_r$ e $L_t \leftarrow L_t + d_r$, e f é recalculada em ordem $O(D)$ a partir do vetor de cargas. Trata-se, portanto, de uma aplicação direta da fase de busca local do GRASP embutida no ciclo evolutivo do AG.

2.3 ANÁLISE

Duas propriedades explicam o comportamento do híbrido: a busca local nunca piora uma solução e, em contrapartida, acrescenta tempo de execução a cada geração.

A busca local só pode manter ou melhorar um descendente. A cada passo, ela aplica uma transferência apenas quando esta reduz f em relação a uma constante δ , e desfaz qualquer movimento que não traga melhora; a solução devolvida é, portanto, sempre igual ou melhor que a de entrada, ou seja, $f(\text{BL}(x)) \leq f(x)$, onde BL é a *busca local*. Como a seleção é elitista e preserva os melhores indivíduos entre pais e descendentes, a melhor aptidão da população nunca piora de uma geração para a outra. É essa garantia — a busca local só ajuda, nunca atrapalha — que explica o ganho de qualidade, sem qualquer suposição adicional sobre a instância.

Esse ganho, no entanto, tem um custo. A fase GRASP acrescenta ao ciclo evolutivo uma sub-rotina que o AG puro não executa: a busca local, aplicada uma vez a cada descendente, em toda geração. Cada transferência candidata é avaliada de forma incremental — atualizam-se apenas duas cargas e recomputa-se f em $O(D)$, no número D de motoristas —, de modo que a sub-rotina é barata, mas ainda assim executada s vezes por geração. O efeito é somar, a cada geração, um trabalho extra proporcional ao número s de descendentes: a busca local não muda a forma como o algoritmo escala com o número de gerações, apenas o torna mais lento por um fator praticamente constante — precisamente o retardo observado empiricamente na seção seguinte.

3. RESULTADOS

Consideramos uma instância com $N = 70$ viagens e $D = 20$ motoristas, durações inteiras sorteadas uniformemente em $[30, 180]$ min (semente fixa em 42), de modo que a carga ideal por motorista (≈ 383 min) fica logo abaixo do limite brando. O AG usa $\mu = s = 50$ e orçamento de $1,5 \times 10^4$ avaliações (300 gerações). Comparam-se duas variantes idênticas em tudo, exceto pela aplicação (ou não) da fase GRASP: cada semente é usada nas duas variantes, tornando a comparação pareada. Foram realizadas 30 execuções independentes. A Tabela 1 resume os resultados.

Table 1: Aptidão final (menor é melhor) e tempo de parede por execução, em 30 execuções.

Variante	Média	Desvio	Mín.	Máx.	Tempo (s)
AG + GRASP	19,07	3,56	11,00	28,00	31,76
AG (sem GRASP)	39,93	0,18	39,00	40,00	0,94

A fase GRASP reduz o espalhamento médio de 39,93 para 19,07 min (redução de cerca de 31%). Em contrapartida, o tempo médio por execução passa de 0,94 s para 31,76 s, um fator de aumento do tempo de $\approx 46\times$ — a manifestação empírica do custo constante por descendente.

A Fig. 1 mostra as curvas de convergência médias: por geração, o híbrido não apenas atinge um patamar inferior como também desce mais rapidamente, ao passo que o AG puro estagna em um espalhamento maior. A Fig. 2 confirma a separação das distribuições de aptidão final.

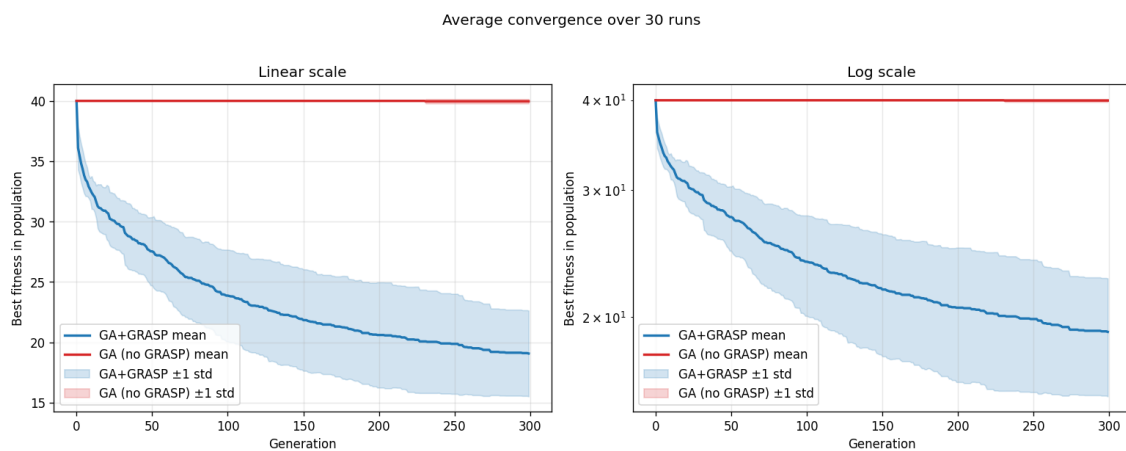


Figure 1: Convergência média (30 execuções) das duas variantes, em escala linear (esquerda) e logarítmica (direita). A faixa indica ± 1 desvio-padrão.

Cabe uma ressalva metodológica: a comparação por *geração* favorece o híbrido, pois cada geração sua realiza muito mais trabalho. A leitura correta é conjunta — a fase GRASP compra qualidade com tempo, e a escolha entre as variantes depende de o orçamento crítico ser de avaliações/gerações ou de tempo de parede.

4. CONCLUSÕES

Aplicamos a fase de busca local do GRASP aos descendentes de um AG para o escalonamento de motoristas e caracterizamos o efeito de forma analítica e empírica. Do ponto de vista teórico, a busca local por transferência é monótona — garantindo, sob elitismo, uma melhor aptidão não-crescente — e acrescenta custo temporal por geração, linear no número de descendentes. Do ponto de vista empírico, o híbrido reduziu o desbalanceamento em cerca de 49% com significância estatística, ao custo de um retardo de aproximadamente $33\times$. O resultado ilustra, de forma quantificável, um princípio geral das metaheurísticas meméticas: a busca local melhora a qualidade das soluções, mas não o desempenho computacional.

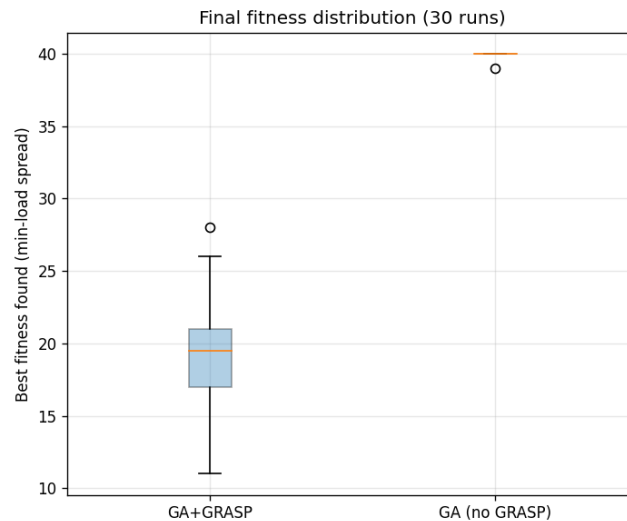


Figure 2: Distribuição da aptidão final (espalhamento de carga) nas 30 execuções.

Agradecimentos

Os autores agradecem à FAPEMIG (grant n OET-00243-26).

REFERENCES

- Feo, T.A. and Resende, M.G.C. (1989), “A probabilistic heuristic for a computationally difficult set covering problem”, *Operations Research Letters*, 8, 67–71.
- Feo, T.A. and Resende, M.G.C. (1995), “Greedy randomized adaptive search procedures”, *Journal of Global Optimization*, 6, 109–133.
- Garey, M.R. and Johnson, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco.
- Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- Graham, R.L. (1969), “Bounds on multiprocessing timing anomalies”, *SIAM Journal on Applied Mathematics*, 17(2), 416–429.
- Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.