

Resolução do Problema de Alocação de Turnos em Call Centers via Metaheurística GRASP

Ayanne Caruline Gomes Gouveia RA 299939

Beatriz Cardoso Nascimento RA 247403

14 de novembro de 2025

Resumo

Este trabalho abordará o problema de atribuição de turnos com prioridade por senioridade, um desafio clássico de otimização de pessoal. Utilizando o caso de referência da New Brunswick Telephone Company (NBTEL), propomos uma metaheurística GRASP (Greedy Randomized Adaptive Search Procedure) para gerar escalas de trabalho de alta qualidade. Compararemos o desempenho da nossa abordagem com uma heurística baseada em Programação Linear Inteira, que serve como nosso baseline. A avaliação dos resultados será conduzida em múltiplas dimensões, incluindo a qualidade da solução, medida pela satisfação dos funcionários respeitando a hierarquia de senioridade, e a eficiência computacional. Com os experimentos, que foram realizados tanto na instância original do problema quanto em cenários sintéticos, buscamos demonstrar o potencial do GRASP para encontrar soluções superiores e mais robustas para problemas de alocação com objetivos hierárquicos.

1 Introdução

A atribuição de turnos em ambientes de atendimento, como call centers e centrais de suporte, constitui um dos problemas centrais de otimização de força de trabalho na ciência de serviços. Trata-se de uma classe de problemas marcada pela presença simultânea de múltiplas restrições, incluindo disponibilidade individual, exigências de habilidades, limites de jornada, preferências de horário e prioridades organizacionais. Essa combinação de fatores confere ao problema uma natureza altamente combinatória e frequentemente intratável por métodos exatos em instâncias de porte realista, tornando o desenvolvimento de heurísticas e metaheurísticas essencial para aplicações operacionais.

O presente estudo parte do problema clássico da New Brunswick Telephone Company (NBTEL), originalmente formulado por Thompson [1], cuja principal característica é a estrutura de prioridades estritamente preemptivas: em primeiro nível, maximizar a cobertura dos turnos; nos níveis subsequentes, minimizar a insatisfação dos funcionários em ordem estrita de senioridade. Essa hierarquia reflete práticas operacionais reais, nas quais a experiência dos operadores são critérios determinantes na distribuição das cargas de trabalho.

Com base nessa motivação, este trabalho desenvolve e avalia uma versão especializada da metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [2], combinando um processo construtivo guloso-randomizado fundamentado em uma Lista Restrita de Candidatos (RCL) com mecanismos de busca local projetados para operar diretamente sobre uma função objetivo lexicográfica. Essa formulação permite representar de forma exata a estrutura preemptiva sem recorrer a ponderações artificiais, integrando o mecanismo de priorização ao processo heurístico de forma natural e computacionalmente eficiente.

Além da variante tradicional do GRASP, o estudo investiga duas extensões: o *Random Plus Greedy* (RPG), que incorpora uma etapa inicial estocástica com o objetivo de diversificar as soluções construídas, e o *Reactive GRASP*, que ajusta dinamicamente o parâmetro de controle α com base no desempenho histórico das iterações. Os experimentos foram conduzidos tanto sobre a instância original da NBTEL quanto sobre um conjunto de instâncias sintéticas de maior porte, permitindo avaliar a escalabilidade e a robustez das abordagens.

Os resultados obtidos demonstram que o *Reactive GRASP* apresenta desempenho consistentemente superior, produzindo soluções de menor insatisfação, com cobertura elevada e em tempos significativamente inferiores às

demais variantes. Em contrapartida, o solver Gurobi, embora capaz de resolver exatamente o modelo lexicográfico, alcançou o limite de tempo em diversas instâncias ampliadas, evidenciando a dificuldade de escalar métodos exatos para problemas dessa natureza. O GRASP Tradicional apresentou desempenho intermediário, enquanto o RPG, apesar de promover maior diversificação no processo construtivo, produziu soluções de menor qualidade e tempos computacionais substancialmente maiores.

Esses achados demonstram que abordagens metaheurísticas adaptativas, como o *Reactive GRASP*, são especialmente adequadas para problemas com funções objetivo lexicográficas e múltiplas restrições operacionais. Em síntese, o presente trabalho propõe, valida e analisa uma metodologia heurística capaz de preservar com rigor a lógica preemptiva de Thompson, oferecendo simultaneamente uma solução escalável, eficiente e alinhada às necessidades práticas de sistemas reais de escalonamento de turnos.

2 Revisão Bibliográfica

O dimensionamento e a alocação da força de trabalho constituem temas centrais na Pesquisa Operacional, com aplicações diretas em diversos setores de serviços. Em ambientes como centrais de atendimento (*call centers*), a elaboração de escalas eficientes é essencial para equilibrar custos operacionais, assegurar níveis de serviço e preservar a satisfação dos funcionários, fatores que influenciam diretamente a qualidade do atendimento e a rotatividade laboral [3, 4]. A literatura especializada trata esse conjunto de problemas em diferentes níveis de planejamento, que vão do estratégico ao operacional.

No nível estratégico, destaca-se o problema de *staffing*, que consiste em determinar o número ideal de operadores e suas competências de modo a atender uma demanda estocástica e variável. Esse desafio é frequentemente abordado por meio de modelos baseados em simulação e técnicas de otimização para avaliar políticas de contratação e atribuição. Cezik et al. [5] e Atlason et al. [6] são referências importantes nesse contexto, ao combinarem programação linear, planos de corte e simulação para analisar sistemas *single skill* e *multi skill*, evidenciando a interdependência entre decisões de capacidade e níveis de serviço.

Em nível tático, o foco desloca-se para a otimização de contratos e padrões de turnos. Nessa etapa, o objetivo é selecionar estruturas de jornada que atendam de forma eficiente à demanda prevista. Cordone et al. [7, 3] analisam esse problema e demonstram a eficácia da metaheurística GRASP no tratamento de instâncias de grande porte. Esses estudos reforçam o potencial de técnicas heurísticas e metaheurísticas para lidar com problemas complexos de escalonamento.

No nível operacional, encontra-se o problema de atribuição de turnos (*shift assignment* ou *rostering*), no qual o conjunto de funcionários e de turnos já está definido. O caso seminal da New Brunswick Telephone Company (NBTEL), formulado por Thompson [1], tornou-se referência por combinar restrições operacionais rígidas com uma função objetivo estruturada por prioridades estritamente preemptivas. Nesse modelo, minimizar turnos não cobertos constitui o objetivo principal, seguido pela minimização da insatisfação dos funcionários, avaliada em ordem rigorosa de senioridade. Essa estrutura hierárquica inviabiliza a redução do problema a um único objetivo escalar, exigindo mecanismos capazes de preservar relações de prioridade rígidas.

Dada sua complexidade, o problema motivou o desenvolvimento de abordagens heurísticas especializadas. Hojati [8] propôs um método híbrido baseado em Programação Linear Inteira (PLI), composto por duas fases sequenciais: a primeira foca em viabilidade global, minimizando o número de turnos não alocados; a segunda otimiza a alocação individual de cada funcionário, seguindo estritamente a ordem de senioridade. Essa abordagem tornou-se um marco na literatura por oferecer soluções de alta qualidade preservando o caráter preemptivo da função objetivo.

Apesar desses avanços, observa-se que ainda há uma lacuna importante na literatura: poucos estudos investigam o uso de metaheurísticas generalistas em cenários com funções objetivo estritamente preemptivas ou fortemente lexicográficas. Trabalhos existentes concentram-se predominantemente em formulações exatas ou heurísticas altamente específicas, deixando pouco explorado o potencial de métodos adaptativos capazes de equilibrar diversificação e intensificação de forma sistemática.

O presente estudo busca preencher essa lacuna ao propor o desenvolvimento de uma metaheurística GRASP especialmente estruturada para o problema da NBTEL. A proposta integra mecanismos construtivos e de busca local projetados para operar diretamente sobre uma função objetivo lexicográfica, preservando a prioridade preemptiva entre cobertura de turnos e insatisfação por senioridade. Além disso, investiga-se o impacto de estratégias alternativas de construção, incluindo o *Random Plus Greedy* (RPG) e o *Reactive GRASP*, ampliando o entendimento sobre como diferentes políticas de controle de aleatoriedade influenciam a qualidade e a eficiência das soluções.

Ao conduzir uma análise comparativa abrangente entre as variantes do GRASP e métodos exatos, este trabalho contribui para o avanço do estado da arte ao demonstrar que metaheurísticas adaptativas podem lidar de forma eficiente com funções objetivo lexicográficas complexas, oferecendo soluções competitivas para sistemas reais de escalonamento sujeitos a múltiplas restrições operacionais.

3 Descrição do Problema: Thompson (1997)

Nesta seção apresenta-se o problema de atribuição de turnos da NBTel, conforme descrito por Thompson [1]. Embora o autor não forneça uma formulação completa de Programação Linear Inteira (PLI), o artigo apresenta um conjunto estruturado de regras operacionais, prioridades preemptivas e critérios de insatisfação, que podem ser formalizados de maneira equivalente para fins de análise e comparação. O modelo matemático a seguir organiza, de forma explícita, os componentes centrais do problema, tal como especificados pelo autor.

Conjuntos

- S : conjunto de turnos da semana;
- E : conjunto de funcionários, ordenado do mais sênior ($e = 1$) ao menos sênior;
- C_e : conjunto de categorias de turno que o funcionário e está habilitado a trabalhar;
- D_e : conjunto de dias nos quais o funcionário e está disponível.

Variáveis de decisão

- $x_{es} = 1$ se o funcionário e for alocado ao turno s , e 0 caso contrário;
- $u_s = 1$ se o turno s não for coberto, e 0 caso contrário;
- $y_e = 1$ se o funcionário e atingir sua carga máxima semanal de turnos, e 0 caso contrário.

Parâmetros

- m_e : número máximo de turnos que o funcionário e pode trabalhar na semana;
- c_s : categoria do turno s ;
- d_s : dia da semana ao qual o turno s pertence;
- ST_e : horário de início desejado pelo funcionário e ;
- ST_s : horário de início do turno s ;
- ES_e : coeficiente de insatisfação por começar mais cedo que o desejado;
- LS_e : coeficiente de insatisfação por começar mais tarde que o desejado.

Cálculo da insatisfação

Em Thompson [1], o custo de insatisfação não é um parâmetro arbitrário, mas uma função definida a partir da diferença entre o horário desejado e o horário do turno:

$$v_{es} = \begin{cases} ES_e \cdot (ST_e - ST_s), & \text{se } ST_s < ST_e, \\ LS_e \cdot (ST_s - ST_e), & \text{se } ST_s > ST_e, \\ 0, & \text{caso contrário.} \end{cases}$$

Esse custo representa o desconforto associado a iniciar um turno mais cedo ou mais tarde que o horário preferido e é fundamental no modelo original.

Função objetivo preemptiva

O problema apresenta prioridades estritamente preemptivas:

1. prioridade máxima: minimizar turnos não alocados;
2. prioridades subsequentes: minimizar a insatisfação dos funcionários em ordem de senioridade.

Uma forma de formalizar essa estrutura é:

$$\min Z = P_0 \left(\sum_{s \in S} u_s \right) + \sum_{e \in E} P_e \left(\sum_{\{s \in S: c_s \in C_e, d_s \in D_e\}} v_{es} x_{es} \right), \quad (1)$$

em que $P_e \gg P_{e+1}$ para todos os e . Uma unidade de melhoria em um nível é infinitamente mais importante do que qualquer melhoria possível nos níveis inferiores.

Restrições

- **Cobertura ou indicação de turno não alocado:**

$$\sum_{\{e \in E: c_s \in C_e, d_s \in D_e\}} x_{es} + u_s = 1, \quad \forall s \in S.$$

- **Cada funcionário deve receber pelo menos um turno:**

$$\sum_{\{s \in S: c_s \in C_e, d_s \in D_e\}} x_{es} \geq 1, \quad \forall e \in E.$$

- **Limite máximo de turnos por funcionário:**

$$\sum_{\{s \in S: c_s \in C_e, d_s \in D_e\}} x_{es} \leq m_e, \quad \forall e \in E.$$

- **No máximo um turno por dia por funcionário:**

$$\sum_{\{s \in S: c_s \in C_e, d_s = i\}} x_{es} \leq 1, \quad \forall e \in E, \forall i \in D_e.$$

- **Regra de senioridade (regra central de Thompson):** um funcionário menos sênior só pode exceder um turno semanal extra se o mais sênior imediatamente anterior já tiver alcançado sua carga máxima:

$$\sum_{\{s \in S: c_s \in C_e, d_s \in D_e\}} x_{es} \leq 1 + (m_e - 1)y_{e-1}, \quad \forall e > 1,$$

$$m_e y_e \leq \sum_{\{s \in S: c_s \in C_e, d_s \in D_e\}} x_{es}, \quad \forall e < |E|.$$

Thompson não resolve o modelo acima como um único problema de PLI. O autor utiliza um procedimento sequencial em que:

- Resolve um subproblema para cada funcionário, em ordem de senioridade;
- Após alocar um funcionário, retira os turnos escolhidos e passa ao próximo;
- Utiliza penalidades e reduções de custo para garantir a viabilidade global, especificamente:
 1. Penalização de turnos que devem ser reservados para fases posteriores;
 2. Redução de custo quando um turno deve necessariamente ser atribuído ao funcionário atual;
 3. Redução adicional quando restam poucos funcionários elegíveis para um turno.

A formulação apresentada nesta seção é, portanto, uma formalização matemática que sintetiza as regras e prioridades especificadas no artigo, preservando sua estrutura preemptiva e seus critérios de insatisfação.

4 Metodologia

Considerando as lacunas identificadas na literatura, em particular a ausência de metaheurísticas generalistas aplicadas a funções objetivo estritamente preemptivas e ao problema da NBTel, esta seção apresenta a metodologia desenvolvida neste estudo. A abordagem proposta foi concebida de modo a preservar a lógica preemptiva do modelo original, empregando uma função objetivo lexicográfica que reproduz, de forma estrita, a ordem de prioridades entre a cobertura de turnos e a minimização da insatisfação segundo a senioridade.

4.1 Modelo exato de referência em Gurobi

A formulação apresentada na Seção 3 foi implementada no solver Gurobi, utilizando um modelo de programação inteira com objetivo multiobjetivo lexicográfico. As variáveis binárias x_{es} , u_s e y_e foram codificadas conforme o modelo, e as prioridades preemptivas foram representadas por meio do recurso `setObjectiveN`, definindo como primeiro nível a minimização de turnos não cobertos e, em seguida, a minimização da insatisfação de cada empregado em ordem de senioridade. Os dados da instância NBTel foram estruturados em dicionários que armazenam horários de início, categorias de turno, limites semanais e dias de indisponibilidade, conforme descrito no artigo original.

4.2 GRASP

A metaheurística GRASP foi desenvolvida para resolver problemas de otimização combinatória, caracterizados por elevada complexidade decorrente da dificuldade de identificar soluções ótimas em espaços de busca extensos. Seu funcionamento baseia-se em duas fases principais [2]:

1. fase construtiva, responsável pela geração de soluções iniciais a partir de uma lista restrita de candidatos (RCL);
2. fase de busca local, voltada ao aprimoramento das soluções construídas.

O GRASP privilegia a obtenção de uma solução inicial promissora, de modo que a busca local atue como fase de refinamento. Enquanto estratégias puramente gulosas selecionam sempre o melhor elemento a cada iteração, o GRASP adota um mecanismo híbrido, escolhendo candidatos a partir de uma lista restrita de forma aleatória e adaptativa, o que introduz variabilidade e amplia o potencial de exploração do espaço de soluções.

Neste trabalho, o GRASP é empregado para o problema de alocação de turnos com senioridade, inspirado em Thompson [1], em que a função objetivo é avaliada de forma estritamente lexicográfica, priorizando a redução do número de turnos não alocados e, em seguida, a minimização da insatisfação dos empregados em ordem de senioridade.

4.3 Lista Restrita de Candidatos (RCL)

Na fase construtiva da metaheurística, a RCL desempenha papel essencial na geração de soluções diversificadas. Seu principal objetivo é introduzir um elemento de aleatoriedade controlada no processo guloso de construção, equilibrando a busca entre soluções alternativas que garantam a exploração de regiões ainda pouco visitadas do espaço de soluções.

Durante a construção de uma solução parcial S , o conjunto de candidatos C contém todos os elementos que podem ser inseridos em S mantendo a viabilidade. Cada candidato $c \in C$ é avaliado segundo um critério de ganho ou custo incremental $g(c)$, representando o impacto de sua inclusão na função objetivo. A RCL é então formada por um subconjunto dos melhores elementos de C , selecionados de acordo com um parâmetro de controle $\alpha \in [0, 1]$:

$$\text{RCL} = \{c \in C \mid g(c) \leq g_{\min} + \alpha(g_{\max} - g_{\min})\}, \quad (2)$$

em que g_{\min} e g_{\max} correspondem, respectivamente, aos melhores e piores valores de $g(c)$ na iteração corrente. O parâmetro α regula o nível de aleatoriedade: valores baixos tendem a priorizar escolhas gulosas, enquanto valores altos favorecem maior diversificação.

De forma alternativa, a RCL pode ser construída por *cardinalidade*, contendo os k elementos mais promissores:

$$\text{RCL} = \{c_1, c_2, \dots, c_k\}, \quad (3)$$

em que k define o tamanho fixo da lista. Em ambos os casos, a escolha final do elemento a ser adicionado à solução é feita de modo aleatório dentro da RCL, assegurando variabilidade entre iterações.

O uso da RCL torna o GRASP mais robusto, evitando convergência prematura e favorecendo uma exploração equilibrada do espaço de busca. Essa característica contribui para a obtenção de soluções de alta qualidade, especialmente em problemas complexos de otimização combinatória, como cobertura de conjuntos, roteamento de veículos e alocação de recursos.

No contexto específico deste trabalho, a RCL é construída a partir dos turnos elegíveis para cada empregado e , já pré-ordenados segundo o custo de insatisfação $v_{e,s}$. Adota-se um esquema de RCL por limiar, com

$$g(s) = v_{e,s},$$

de modo que a lista restrita contenha os turnos com insatisfação até um limite determinado por α . A seleção final do turno a ser alocado é feita de forma aleatória dentro da RCL, preservando o caráter guloso-randomizado do método.

4.4 Modelagem da solução e avaliação lexicográfica

Para cada instância, considera-se o conjunto de empregados E , ordenados do mais sênior para o menos sênior, e o conjunto de ocorrências de turno S , definidos por pares (d, sid) , em que d representa o dia da semana e sid identifica o turno. Cada empregado $e \in E$ possui:

- um conjunto de habilidades C_e ;
- um conjunto de dias disponíveis D_e ;
- uma capacidade máxima de turnos m_e ;
- parâmetros que determinam sua insatisfação em função do horário de início e da posição do intervalo de almoço.

A insatisfação individual $v_{e,s}$ ao atribuir o turno s ao empregado e é calculada a partir do desvio entre o horário desejado e o horário do turno, com possível penalidade associada ao horário de almoço:

$$v_{e,s} = \begin{cases} ES_e \cdot (ST_e - ST_s) + \lambda \cdot |BL_s - LL_e|, & ST_s < ST_e, \\ LS_e \cdot (ST_s - ST_e) + \lambda \cdot |BL_s - LL_e|, & ST_s \geq ST_e, \end{cases}$$

em que ST_e é o horário de início desejado pelo empregado, ST_s é o horário de início do turno, BL_s é o horário de início do intervalo de almoço do turno, LL_e é o horário ideal de almoço do empregado, ES_e e LS_e são coeficientes de insatisfação por antecipar ou postergar o início do turno, e λ é um parâmetro de ponderação.

A qualidade de uma solução S é avaliada por um vetor de custos lexicográfico:

$$\text{comp}(S) = (\# \text{turnos não alocados}, \text{diss}_{E_1}, \text{diss}_{E_2}, \dots, \text{diss}_{E_{|E|}}),$$

em que diss_{E_i} representa a insatisfação total do empregado na i -ésima posição de senioridade. A comparação entre duas soluções S_A e S_B é estritamente lexicográfica:

$$S_A \prec_{\text{lex}} S_B \iff \text{comp}(S_A) < \text{comp}(S_B),$$

comparando as tuplas S_A e S_B item a item, o que garante que a prioridade principal seja sempre a cobertura máxima de turnos e, em seguida, a minimização da insatisfação dos empregados mais seniores.

Para fins de registro numérico e análise, utiliza-se ainda um valor agregado:

$$\text{report_value}(S) = P \cdot \# \text{turnos não alocados} + \sum_{e \in E} \text{diss}_e,$$

com P suficientemente grande ($P = 10^6$), de modo que qualquer unidade de turno não alocado supere, em magnitude, variações de insatisfação.

4.5 Métodos de busca local

Os métodos de busca local constituem um componente essencial em diversas metaheurísticas, incluindo o GRASP, desempenhando o papel de refinamento das soluções construídas na fase inicial. Após a obtenção de uma solução viável, a busca local explora sistematicamente sua vizinhança, em busca de uma configuração que produza uma melhoria na função objetivo.

Formalmente, seja S uma solução corrente e $N(S)$ o conjunto de soluções vizinhas de S , definidas segundo uma estrutura de vizinhança apropriada ao problema em questão (como trocas, inserções, remoções ou inversões). O objetivo da busca local é encontrar $S' \in N(S)$ tal que $f(S') < f(S)$ em problemas de minimização. O processo é repetido iterativamente até que não haja vizinho que produza melhoria, caracterizando um *mínimo local*.

Existem dois esquemas principais de exploração da vizinhança:

- *First Improvement*: a busca percorre a vizinhança de forma sequencial e move-se para o primeiro vizinho que proporcione uma melhoria na função objetivo. Essa estratégia tende a ser mais rápida, pois evita a avaliação completa de $N(S)$;
- *Best Improvement*: avalia-se todo o conjunto $N(S)$, e a solução escolhida é aquela que gera a maior melhoria possível. Embora mais custosa computacionalmente, essa abordagem pode conduzir a soluções de melhor qualidade em algumas instâncias.

A eficiência da busca local depende fortemente da definição da vizinhança e do critério de parada. Vizinhanças muito restritas podem levar a ótimos locais de baixa qualidade, enquanto vizinhanças excessivamente amplas aumentam o custo computacional. Por essa razão, é comum empregar estratégias híbridas, como a combinação de múltiplas vizinhanças ou o uso de técnicas de intensificação e diversificação, a exemplo da *Variable Neighborhood Search* (VNS).

No contexto do GRASP, a busca local é aplicada a cada solução construída, e o melhor resultado entre todas as iterações é armazenado como solução final. Essa integração entre construção gulosa randomizada e refinamento local é um dos fatores que conferem ao GRASP um bom equilíbrio entre exploração e intensificação, e explicam sua robustez em problemas de otimização combinatória.

4.5.1 Busca local proposta para o problema de alocação com senioridade

Neste trabalho, a fase de busca local é composta por duas vizinhanças específicas, aplicadas em esquema *first improvement*:

1. **Seniority-Swap**: são considerados pares de empregados $(e_{\text{sen}}, e_{\text{jun}})$, em que e_{sen} é mais sênior que e_{jun} . Para cada par, avaliam-se trocas de turnos (s_1, s_2) tais que: (i) ambos os empregados sejam elegíveis para os turnos trocados, (ii) as restrições de disponibilidade diária sejam respeitadas e (iii) a insatisfação do empregado mais sênior seja reduzida. A primeira troca que satisfaz essas condições é imediatamente aceita;
2. **Uncovered-Shift Relocation**: enquanto existirem turnos não alocados, tenta-se atribuir um desses turnos a algum empregado ainda com capacidade disponível, priorizando os menos seniores e selecionando, dentre estes, aquele que apresenta menor insatisfação para o turno considerado.

As duas vizinhanças são exploradas de forma intercalada até que nenhuma melhoria adicional seja encontrada ou seja atingido um limite máximo de iterações internas da busca local.

4.6 Estratégias alternativas de construção

Além da construção padrão descrita anteriormente, este trabalho implementa e compara duas estratégias alternativas de construção, a fim de avaliar o impacto de diferentes abordagens na qualidade e na diversidade das soluções obtidas.

4.6.1 Random Plus Greedy (RPG)

A estratégia *Random Plus Greedy* (RPG) consiste em uma variação híbrida do processo construtivo tradicional, combinando uma fase puramente aleatória com uma fase orientada por critérios gulosos. O objetivo é explorar

de forma mais ampla o espaço de soluções viáveis, preservando, contudo, a eficiência característica dos métodos construtivos determinísticos.

Na primeira etapa, uma parcela das decisões de alocação é tomada de forma puramente aleatória entre os candidatos viáveis, introduzindo diversidade e reduzindo a dependência do processo em relação às condições iniciais. Em seguida, a solução parcial resultante é completada por meio da heurística gulosa randomizada baseada na RCL, que prioriza a inclusão dos elementos com maior contribuição marginal segundo a função objetivo.

Essa combinação busca equilibrar exploração (pela aleatoriedade controlada) e intensificação (pela inserção orientada dos elementos mais promissores). O RPG é particularmente útil em instâncias nas quais a construção puramente gulosa tende a produzir soluções similares ou restritas a uma mesma região do espaço de busca. Ao incorporar um componente estocástico, essa estratégia amplia o conjunto de soluções candidatas fornecidas à busca local subsequente, aumentando a probabilidade de atingir regiões de melhor qualidade no espaço de soluções.

4.6.2 Reactive GRASP

O *Reactive GRASP* representa uma extensão adaptativa do método clássico, cujo objetivo é ajustar dinamicamente o parâmetro de controle α utilizado na fase construtiva. Enquanto o GRASP tradicional adota um valor fixo de α ao longo de todas as iterações, o Reactive GRASP introduz um mecanismo de aprendizado que modifica esse valor conforme o desempenho observado durante a execução.

A motivação dessa abordagem reside no fato de que a qualidade das soluções geradas pode variar significativamente em função do parâmetro α , que regula o equilíbrio entre comportamento guloso e aleatoriedade. Em vez de se pré-definir um único valor, o Reactive GRASP mantém um conjunto discreto de possíveis valores $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$ e associa a cada um uma probabilidade inicial de seleção idêntica.

Após cada iteração, a probabilidade de escolha de cada α_i é atualizada com base na qualidade das melhores soluções obtidas sob aquele parâmetro. Assim, valores de α que conduzem a soluções mais promissoras passam a ter maior probabilidade de serem selecionados nas iterações seguintes. Esse processo de retroalimentação constitui uma forma simples, mas eficaz, de aprendizado adaptativo, permitindo que o algoritmo ajuste seu comportamento às características específicas do problema tratado.

Se $Q(\alpha_i)$ representa uma medida de qualidade associada ao melhor valor de função objetivo já alcançado com o parâmetro α_i , a nova probabilidade p_i pode ser atualizada segundo:

$$p_i = \frac{Q(\alpha_i)}{\sum_{j=1}^m Q(\alpha_j)}, \quad i = 1, \dots, m. \quad (4)$$

Esse mecanismo garante que o algoritmo privilegie regiões do espaço de busca que apresentem melhor desempenho, sem eliminar completamente a exploração de alternativas. A natureza adaptativa do Reactive GRASP o torna especialmente útil em problemas nos quais o comportamento ótimo de α é fortemente dependente da instância ou do estágio da busca.

Em síntese, o Reactive GRASP aprimora o equilíbrio entre intensificação e diversificação de maneira autônoma, reduzindo a necessidade de calibração manual e aumentando a robustez do método frente a diferentes tipos de problema.

4.7 Execução em lote e coleta de resultados

A implementação em `Python` foi estruturada de modo a permitir a execução em lote de todas as instâncias disponíveis em um diretório. Para cada arquivo de instância, são extraídos os dados de empregados, turnos e exigências de demanda, e o GRASP é executado nas três variantes (tradicional, RPG e Reactive).

Ao final de cada execução, são registradas em um arquivo `.csv` as seguintes métricas para cada combinação de instância e estratégia:

- Número de turnos cobertos e não cobertos;
- Insatisfação média por empregado;
- Tempo total de execução;
- Número total de iterações realizadas dentro do limite de tempo.

Essas informações permitem comparar sistematicamente o desempenho das diferentes estratégias de construção e analisar o comportamento da metaheurística sob diversas instâncias.

4.8 Critérios de parada

Foram utilizados três tipos de critérios de parada:

- **Critérios globais (GRASP):** o laço principal de iterações do GRASP é controlado por um limite de tempo máximo por execução, denotado por t_{\max} (neste trabalho, $t_{\max} = 1800$ segundos). As iterações são executadas enquanto $t < t_{\max}$, registrando-se sempre a melhor solução encontrada até o momento. Além disso, também existe um critério de número máximo de iterações (500) sem melhoria na função objetivo ;
- **Critérios internos (busca local):**
 1. a busca local é encerrada quando nenhuma das vizinhanças (Seniority-Swap e Uncovered-Shift Relocation) consegue produzir uma melhoria na solução corrente;
 2. adicionalmente, é imposto um limite máximo de iterações internas da busca local, a fim de evitar tempos excessivos em instâncias mais difíceis.

Essa combinação de critérios assegura que o procedimento respeite um orçamento de tempo predefinido, ao mesmo tempo em que explora intensamente a vizinhança de soluções promissoras, conforme requerido em aplicações práticas de otimização combinatória.

4.9 Reprodutibilidade e repositório público

Para reprodutibilidade, foram disponibilizados (i) o código-fonte da implementação (GRASP, RPG e Reactive), (ii) geradores de instâncias compatíveis com [1], (iii) scripts de calibração e (iv) coleções de arquivos `.py` com instâncias de tamanhos pequeno, médio e grande, além das sementes fixas utilizadas nos experimentos.

Repositório: <https://github.com/beacnascimento/nbtel-optimization>

5 Resultados e Discussão

Esta seção apresenta a análise empírica dos métodos desenvolvidos, com foco na comparação sistemática entre o modelo exato multiobjetivo e as variantes da metaheurística GRASP propostas. O objetivo é avaliar a qualidade das soluções obtidas, a eficiência computacional dos métodos e seu comportamento sob diferentes níveis de complexidade estrutural. Para isso, realizam-se experimentos controlados em um ambiente computacional padronizado e sobre um conjunto de instâncias que inclui tanto o caso clássico da NBTel quanto cenários sintéticos de maior escala. A discussão dos resultados está organizada em três partes: 5.1 Configuração Experimental, 5.2 Instâncias de Test e 5.3 Resultados e Análises.

5.1 Configuração Experimental

Os experimentos computacionais foram conduzidos no seguinte ambiente:

- **Hardware:**
 - Processador (CPU): AMD Ryzen 7 7730U, 8 núcleos físicos e 16 threads.
- **Sistema operacional:**
 - Ubuntu 22.04.5 LTS.
- **Software:**
 - Solver de otimização: Gurobi Optimizer, versão 12.0.3.
 - Linguagem de programação: Python.

5.2 Instâncias de Teste

Para complementar as instâncias de benchmark da literatura [1], que representa um cenário de menor escala, foi desenvolvido um gerador paramétrico de instâncias. Este gerador permitiu a criação de cenários de teste customizados, visando avaliar a escalabilidade dos métodos e sua sensibilidade a diferentes níveis de restrição.

Todos os scripts de geração estão disponíveis no repositório público. A seguir, detalha-se a parametrização utilizada para os conjuntos de instâncias geradas.

O gerador cria os três dicionários de dados (`employee_data`, `shift_data`, `shift_requirements`) com base em um conjunto de parâmetros de entrada. A lógica de geração segue os seguintes passos:

1. **Definição da Demanda:** Primeiramente, define-se a escala do problema, incluindo o número de dias ($N_DAYS = 7$), o número de funcionários ($N_EMPLOYEES$) e a demanda de turnos por dia ($N_SHIFTS_PER_DAY$).
2. **Geração de Habilidades (Skills):** Para simular um cenário realista, as habilidades (TT) não são uniformemente distribuídas. O gerador utiliza um sistema de pesos (`skill_weights`), onde habilidades “comuns” (ex: IDs 1-5) têm maior probabilidade de serem exigidas pelos turnos do que habilidades “raras”.
3. **Criação de Funcionários:** Cada funcionário (E-i) é criado com um conjunto aleatório de parâmetros, incluindo:
 - Habilidades (SklTyp): Um número de habilidades entre `MIN_SKILLS_PER_EMP` e `MAX_SKILLS_PER_EMP`.
 - Disponibilidade (UnDay): Um número de dias de indisponibilidade sorteado entre 0 e `MAX_UNAVAILABLE_DAYS`.
 - Carga de Trabalho (MxWk): A carga máxima semanal é definida entre `MIN_MXWK` e `MAX_MXWK`.
 - Preferências: Preferências de horário de início (ST), almoço (LL) e custos de penalidade (ES, LS) são sorteados aleatoriamente.
4. **Criação da Demanda de Turnos:** Para cada dia da semana, o gerador cria $N_SHIFTS_PER_DAY$ ocorrências de turno. A cada ocorrência é atribuído um ID único, e seus dados (início ST, almoço BL e habilidade exigida TT) são gerados aleatoriamente, respeitando a distribuição de pesos das habilidades.

Para avaliar o desempenho e a escalabilidade dos algoritmos, foi gerado um conjunto de quatro instâncias sintéticas com dimensões variadas. Três instâncias (120 funcionários/40 turnos, 135/45, e 180/60) foram projetadas para testar diretamente o impacto do aumento de volume. Uma quarta instância, de maior porte (250/90), foi introduzida para avaliar um cenário com uma proporção diferente. A complexidade do problema também foi variada: enquanto três instâncias operaram com um *pool* de 8 *skills*, a instância de 180 funcionários aumentou a complexidade de alocação ao considerar 15 *skills* distintas. Em todos os cenários, as restrições individuais foram mantidas rigorosas, com funcionários possuindo entre 2 e 5 *skills* (reduzido para 2 e 4, no caso de 15 *skills*) e um máximo de 1 dia de indisponibilidade.

5.3 Resultados e Análises

Esta subseção apresenta a avaliação quantitativa e qualitativa das três variantes da metaheurística desenvolvidas neste estudo: *GRASP Tradicional*, *Random Plus Greedy* (RPG) e *Reactive GRASP*. Os resultados são comparados com o modelo exato multiobjetivo lexicográfico resolvido pelo Gurobi, utilizado como referência. A análise segue os critérios definidos na metodologia: (i) qualidade das soluções, medida pela insatisfação média por empregado; (ii) capacidade de cobertura dos turnos; e (iii) desempenho computacional, avaliado pelo tempo de execução e pelo número de iterações.

Os experimentos foram conduzidos em cinco instâncias representativas, incluindo a instância Baseline, correspondente ao problema original da NBTel, e quatro instâncias ampliadas que apresentam diferentes tamanhos e graus de restrição.

Desempenho em termos de qualidade da solução

A Tabela 1 e a Figura 1 evidenciam diferenças marcantes na qualidade das soluções obtidas pelas estratégias avaliadas, medida pela insatisfação média por empregado. O *Gurobi* apresenta, de forma consistente, os menores valores em todas as instâncias, constituindo o limite ótimo para comparação.

Tabela 1: Insatisfação Média por Funcionário

| Instância | Gurobi | GRASP Tradicional | GRASP RPG | Reactive GRASP |
|---------------|-------------|-------------------|-----------|----------------|
| Baseline | 19.6 | 19.6 | 33.1 | 16.4 |
| G1_630 turnos | 2.7 | 29.6 | 48.6 | 15.7 |
| G2_315 turnos | 16.4 | 25.7 | 41.1 | 20.1 |
| G3_280 turnos | 17.8 | 26.6 | 45.9 | 20.6 |
| G4_420 turnos | 24.3 | 39.9 | 47.7 | 31.4 |

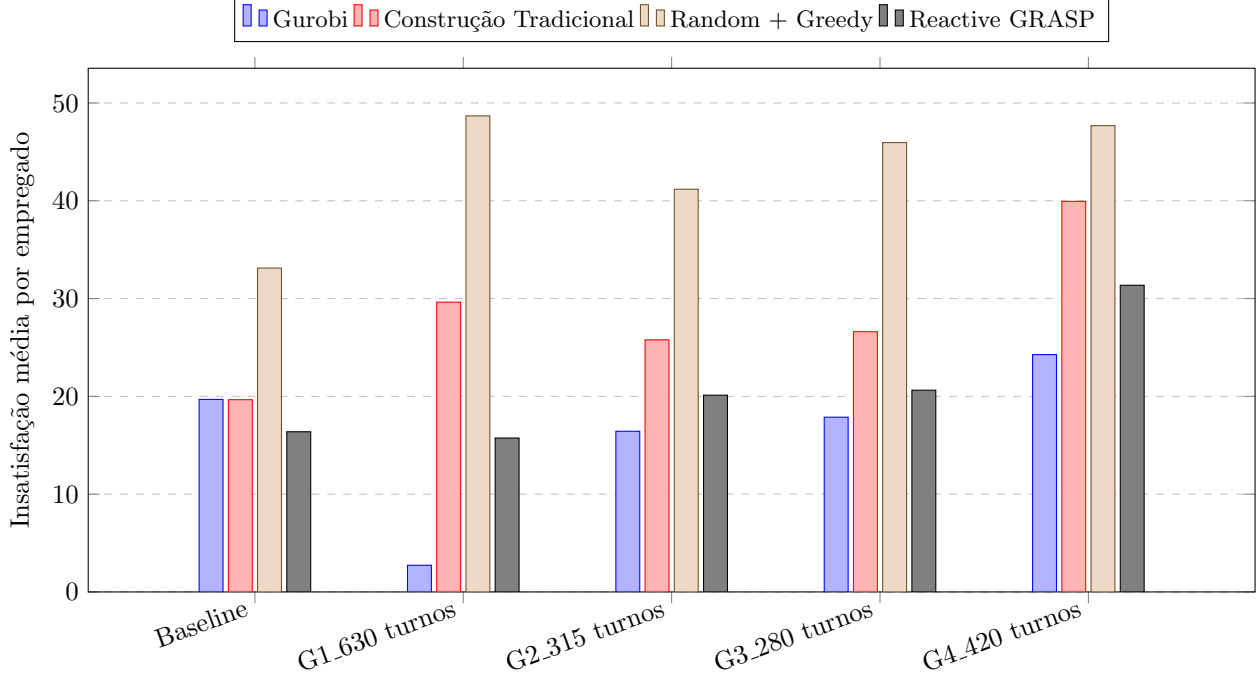


Figura 1: Comparação da insatisfação média por empregado entre estratégias de solução.

Entre as metaheurísticas, o *Reactive GRASP* destaca-se como a abordagem mais eficaz, com valores de insatisfação próximos aos do método exato ao longo das instâncias analisadas. Seu mecanismo adaptativo de seleção de α permite um equilíbrio superior entre diversificação e intensificação, refletido nos resultados apresentados.

O *GRASP Tradicional* exibe desempenho intermediário, com aumento significativo da insatisfação em instâncias maiores, enquanto a estratégia *Random + Greedy* mostra-se claramente a menos eficiente, apresentando os maiores valores na Tabela 1 e na Figura 1.

De forma sintética, os resultados confirmam o seguinte ordenamento de desempenho:

$$\text{Gurobi} > \text{Reactive GRASP} > \text{GRASP Tradicional} > \text{RPG}.$$

Assim, o *Reactive GRASP* configura-se como a melhor alternativa heurística, alcançando soluções de alta qualidade com custo computacional reduzido.

Cobertura de turnos

A cobertura de turnos, correspondente ao nível primordial da função objetivo preemptiva, constitui o critério central para avaliar a viabilidade das soluções geradas. Os resultados evidenciam que todas as variantes da metaheurística atingiram 100% de cobertura nas quatro instâncias de maior porte, o que confirma a capacidade dos procedimentos construtivos e das vizinhanças de busca local em garantir escalas integralmente factíveis mesmo sob maior complexidade combinatória.

Na instância *Baseline*, caracterizada por uma estrutura mais restritiva e menor flexibilidade de alocação, observou-se a única ocorrência de cobertura parcial. Ainda assim, o *Reactive GRASP* apresentou o melhor de-

Tabela 2: Turnos Cobertos (%)

| Instância | Gurobi | GRASP Tradicional | GRASP RPG | Reactive GRASP |
|---------------|--------|-------------------|-----------|----------------|
| Baseline | 100% | 95.39% | 94.68% | 97.87% |
| G1.630 turnos | 100% | 100% | 100% | 100% |
| G2.315 turnos | 100% | 100% | 100% | 100% |
| G3.280 turnos | 100% | 100% | 100% | 100% |
| G4.420 turnos | 100% | 100% | 100% | 100% |

sempenho, alcançando 97.87% de cobertura, seguido pelo GRASP Tradicional (95.39%) e pelo RPG (94.68%). Esses resultados reforçam que o mecanismo adaptativo do Reactive promove escolhas construtivas mais adequadas para contextos de maior rigidez, enquanto o componente aleatório inicial do RPG tende a conduzir a soluções que preservam menos turnos viáveis ao longo da construção.

Desempenho computacional

Tabela 3: Tempo de Execução (s)

| Instância | Gurobi | GRASP Tradicional | GRASP RPG | Reactive GRASP |
|---------------|---------|-------------------|-----------|----------------|
| Baseline | 30.60 | 30.97 | 374.69 | 5.80 |
| G1.630 turnos | 1800.02 | 1801.29 | 1164.49 | 21.83 |
| G2.315 turnos | 937.98 | 575.73 | 371.42 | 18.77 |
| G3.280 turnos | 267.47 | 394.75 | 403.27 | 5.06 |
| G4.420 turnos | 1177.51 | 1034.61 | 1256.89 | 19.72 |

A Tabela 3 apresenta os tempos de execução das diferentes estratégias e evidencia contrastes significativos em termos de custo computacional. Observa-se que o *Reactive GRASP* foi, de forma consistente, a heurística mais eficiente, obtendo os menores tempos em todas as instâncias. Mesmo em casos de maior escala, seus tempos permaneceram na ordem de poucos segundos ou dezenas de segundos, destacando sua elevada eficiência prática.

O *Gurobi*, embora produza soluções de excelente qualidade, apresentou tempos elevados nas instâncias maiores, refletindo o aumento exponencial do esforço computacional requerido pelo método exato. As estratégias *GRASP Tradicional* e *RPG* exibiram tempos substancialmente mais altos que o *Reactive GRASP*, especialmente nas instâncias G1.630 e G4.420, indicando menor eficiência computacional.

De maneira geral, os resultados demonstram que:

Reactive GRASP é a estratégia mais rápida, seguida por Gurobi, GRASP Tradicional e, por fim, RPG.

Assim, o *Reactive GRASP* destaca-se não apenas pela qualidade das soluções, mas também pelo desempenho computacional superior.

Síntese Comparativa das Variantes do GRASP

A Tabela 4 sintetiza o desempenho das três variantes do GRASP em termos de qualidade da solução, cobertura, tempo de execução e número de iterações. Os resultados mostram que o *Reactive GRASP* apresenta o melhor equilíbrio entre qualidade e eficiência. Em todas as instâncias, essa abordagem alcança os menores valores de insatisfação média, mantendo cobertura de 100% nas instâncias sintéticas e tempos de execução substancialmente inferiores aos das demais estratégias. Além disso, o número de iterações necessárias é consistentemente menor ou comparável, evidenciando maior eficácia exploratória.

O *GRASP Tradicional* exibe desempenho intermediário: apesar de atingir cobertura total, apresenta valores de insatisfação mais elevados e tempos de execução significativamente maiores, sobretudo nas instâncias de maior porte. Já o *Random + Greedy (RPG)* apresenta o pior desempenho geral, com insatisfações mais altas e tempos de execução elevados, indicando que a fase aleatória inicial compromete tanto a qualidade quanto a eficiência do processo iterativo.

Tabela 4: Resumo dos resultados das variantes do GRASP: qualidade, cobertura, tempo e iterações

| Instância | Estratégia | Insatisfação Média | Cobertura (%) | Tempo (s) | Iterações |
|---------------|-------------|--------------------|---------------|--------------|------------|
| Baseline | Tradicional | 19.66 | 95.39% | 30.97 | 579 |
| | RPG | 33.12 | 94.68% | 374.69 | 791 |
| | Reactive | 16.38 | 97.87% | 5.80 | 565 |
| G1.630 turnos | Tradicional | 29.63 | 100% | 1801.29 | 798 |
| | RPG | 48.68 | 100% | 1164.49 | 552 |
| | Reactive | 15.74 | 100% | 21.83 | 578 |
| G2.315 turnos | Tradicional | 25.78 | 100% | 575.73 | 664 |
| | RPG | 41.18 | 100% | 371.42 | 516 |
| | Reactive | 20.12 | 100% | 18.77 | 1523 |
| G3.280 turnos | Tradicional | 26.61 | 100% | 394.75 | 561 |
| | RPG | 45.95 | 100% | 403.27 | 665 |
| | Reactive | 20.63 | 100% | 5.06 | 509 |
| G4.420 turnos | Tradicional | 39.94 | 100% | 1034.61 | 807 |
| | RPG | 47.68 | 100% | 1256.89 | 1143 |
| | Reactive | 31.36 | 100% | 19.72 | 1074 |

Em síntese, os resultados confirmam que:

Reactive GRASP é a variante mais robusta, eficiente e consistente entre as versões avaliadas.

A estratégia adapta-se melhor às características hierárquicas e combinatórias do problema, produzindo soluções superiores com menor custo computacional.

6 Conclusão e Trabalhos Futuros

Este trabalho investigou o problema de atribuição de turnos com prioridades preemptivas, inspirado no caso clássico da NBTel [1], por meio de uma formulação lexicográfica e de variantes da metaheurística GRASP especialmente adaptadas à estrutura de senioridade. Ao longo do estudo, foi desenvolvido um arcabouço que integra um modelo exato multiobjetivo em Gurobi, procedimentos construtivos guloso-randomizados baseados em listas restritas de candidatos e operadores de busca local desenhados para preservar a hierarquia de prioridades.

Os experimentos realizados, envolvendo a instância original da NBTel e instâncias sintéticas de maior porte, mostraram que abordagens puramente exatas apresentam dificuldades de escalabilidade, enquanto as variantes do GRASP são capazes de produzir soluções viáveis de alta qualidade em tempos significativamente menores. Entre essas variantes, o *Reactive GRASP* destacou-se de forma consistente: obteve menores níveis de insatisfação, manteve cobertura total nas instâncias de maior porte e apresentou tempos de execução de ordem de segundos, conforme discutido na seção de Resultados.

De maneira geral, os achados indicam que a combinação entre modelagem lexicográfica explícita e mecanismos adaptativos de intensificação e diversificação torna o GRASP uma ferramenta apropriada para problemas de escalonamento com prioridades estritas de senioridade. O *Reactive GRASP*, em particular, emerge como a estratégia mais eficiente e robusta entre as avaliadas, oferecendo um compromisso favorável entre qualidade da escala, respeito à hierarquia de preferências e viabilidade computacional em cenários de maior complexidade combinatória. Portanto, a utilização das variantes do GRASP para o problema de alocação de turnos com restrição de senioridade é interessante para *stakeholders* de *call centers* de grande porte, como podemos observar pelo desempenho do *Reactive GRASP*, e se mostra uma área de investigação promissora.

Trabalhos Futuros

Os resultados alcançados abrem diversas possibilidades de extensão e aprofundamento, tanto no plano metodológico quanto no aplicado. Algumas direções promissoras para trabalhos futuros incluem:

- **Aprimoramento das vizinhanças de busca local.** Podem ser exploradas vizinhanças mais complexas, como trocas múltiplas de turnos, movimentos de realocação em cadeia e estruturas inspiradas em *Variable*

Neighborhood Search (VNS), com o objetivo de escapar de mínimos locais mais profundos e explorar regiões adicionais do espaço de soluções.

- **Integração com métodos exatos em esquemas matheurísticos.** Uma linha de pesquisa relevante consiste em combinar o *Reactive GRASP* com modelos de PLI em fases de refinamento ou reparo, utilizando o GRASP para gerar soluções iniciais de alta qualidade e o Gurobi para otimizar subconjuntos de decisões selecionados de forma criteriosa.
- **Extensão para horizontes de planejamento maiores e problemas integrados.** O presente estudo considerou um horizonte semanal fixo. Extensões naturais incluem a modelagem multiperíodo (várias semanas) e a integração com decisões de *staffing* e seleção de contratos, aproximando o problema operacional de atribuição do contexto tático analisado por Cordone et al. [7, 3].
- **Inclusão de novos critérios de equidade e bem-estar.** A modelagem pode ser ampliada para contemplar medidas adicionais de justiça, como balanceamento de turnos noturnos, distribuição de finais de semana trabalhados e limitação de sequências desfavoráveis de turnos, incorporando esses aspectos à função objetivo lexicográfica ou a restrições suplementares.
- **Calibração automática e variantes do esquema reativo.** Embora o *Reactive GRASP* tenha apresentado desempenho robusto, há espaço para investigar outras políticas de atualização de probabilidades, critérios de avaliação agregada e mecanismos de aprendizado adaptativo, incluindo o uso de técnicas de *machine learning* para ajustar parâmetros com base em características das instâncias.
- **Validação em dados reais e estudos de caso industriais.** Uma etapa importante consiste em aplicar o arcabouço desenvolvido a bases de dados reais de centrais de atendimento ou outros ambientes de escalonamento intensivo, avaliando o impacto das soluções propostas em indicadores operacionais, como nível de serviço, satisfação dos funcionários e custo de mão de obra.

De forma geral, as evidências empíricas indicam que a combinação entre modelagem lexicográfica, mecanismos construtivos guloso randomizados e adaptação reativa de parâmetros constitui uma linha promissora para o desenvolvimento de algoritmos de escalonamento flexíveis, escaláveis e alinhados às necessidades práticas de organizações intensivas em mão de obra.

7 Distribuição de Tarefas

Para a realização deste trabalho, as atividades foram distribuídas entre as integrantes da equipe de forma a garantir eficiência e coerência metodológica.

Ayanne Gouveia foi responsável pela pesquisa e seleção dos artigos pertinentes, realizando a análise comparativa e a verificação de equivalência entre as referências identificadas na literatura. Além dessas etapas, ajudou na implementação do algoritmo de Programação Linear Inteira (PLI) com otimização via Gurobi, foi responsável pela implementação integral do código baseado na metaheurística GRASP, abrangendo suas variantes, testes e ajustes necessários. Por fim, assumiu a análise e avaliação crítica dos resultados obtidos, integrando as evidências teóricas e computacionais ao relatório final.

Beatriz Nascimento, por sua vez, encarregou-se da revisão bibliográfica geral, da implementação do repositório no GitHub com o artigo e as instâncias de Thompson [1], da geração de instâncias sintéticas, da implementação do algoritmo PLI com otimização via Gurobi, e da redação da Introdução. Adicionalmente, ela foi responsável pela formatação final do relatório, colaborou na seção de Procedimentos de Comparação e Proposta de uma Metaheurística GRASP para o Problema de Atribuição com Senioridade, e pela descrição do problema de Thompson [1]. Colaborou na implementação do GRASP, na revisão geral do relatório e foi responsável por executar os experimentos.

Referências

- [1] Gary M Thompson. “Assigning telephone operators to shifts at New Brunswick Telephone Company”. Em: *Interfaces* 27.4 (1997), pp. 1–11.

- [2] M. G. C. Resende e C. C. Ribeiro. *Greedy Randomized Adaptive Search Procedures*. Acesso em: 11 set. 2025. 2003. URL: <https://profs.ic.uff.br/~celso/artigos/resende-ribeiro-GRASP-HMH3.pdf>.
- [3] Roberto Cordone et al. “Optimal selection of contracts and work shifts in multi-skill call centers”. Em: *EURO Journal on Computational Optimization* 2.4 (2014), pp. 247–277.
- [4] KO De Ruyter, Martin Wetzels e Richard Feinberg. “Role stress in call centers: Its effects on employee performance and satisfaction”. Em: *Journal of interactive marketing* 15.2 (2001), pp. 23–35.
- [5] Mehmet Tolga Cezik e Pierre L’Ecuyer. “Staffing multiskill call centers via linear programming and simulation”. Em: *Management Science* 54.2 (2008), pp. 310–323.
- [6] Júlíus Atlason, Marina A Epelman e Shane G Henderson. “Call center staffing with simulation and cutting plane methods”. Em: *Annals of Operations Research* 127.1 (2004), pp. 333–358.
- [7] Roberto Cordone et al. “Optimization of multi-skill call centers contracts and work-shifts”. Em: *Service Science* 3.1 (2011), pp. 67–81.
- [8] Mehran Hojati. “Near-optimal solution to an employee assignment problem with seniority”. Em: *Annals of Operations Research* 181.1 (2010), pp. 539–557.