

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA



TRABALHO COMPUTACIONAL

Entrega 2 - Resultados da otimização multiobjetivo

Autores:

GABRIEL ROLLA FERREIRA - 2022038457
LUCAS FRAZÃO MOREIRA - 2020056857
LUCAS PIMENTA BRAGA - 2023034552
MATEUS DE SOUZA GONTIJO - 2020053530

Disciplina:

TEORIA DA DECISÃO

Belo Horizonte

Setembro de 2025

Sumário

1	Introdução e Objetivos	2
2	Definição do Problema	2
3	Formulação Matemática	3
3.1	Função Objetivo 1: Minimização do Custo Total	3
3.2	Função Objetivo 2: Minimização do Desequilíbrio de Carga	3
3.3	Problema Multiobjetivo	4
4	Algoritmo de Solução	4
4.1	Modelagem computacional	4
4.2	Estruturas de vizinhança	4
4.3	Heurística construtiva	4
4.4	Estratégia de refinamento	4
5	Resultados da Otimização Mono-Objetivo	5
5.1	Resultados numéricos	5
5.2	Curvas de convergência	5
5.3	Melhor solução encontrada	6
6	Otimização Multiobjetivo	8
6.1	Abordagem Escalar: Soma Ponderada (Pw)	8
6.2	Abordagem Escalar: Método ϵ -Restrito (PE)	9
6.3	Resultados da Otimização Multiobjetivo	9
6.3.1	Resultados da Soma Ponderada (Pw)	10
6.3.2	Resultados do Método ϵ -Restrito (PE)	10
6.3.3	Comparação das Abordagens	11
7	Conclusões	11
8	Bibliografia de Referência	12

1 Introdução e Objetivos

A atribuição eficiente de tarefas a agentes é um problema clássico de otimização que aparece em diversos contextos práticos, como a distribuição de atividades entre trabalhadores, o escalonamento de processos em máquinas, ou a alocação de recursos em sistemas computacionais. Em todos esses cenários, a má distribuição pode gerar custos elevados ou sobrecarga de determinados agentes, comprometendo o desempenho global do sistema.

Neste trabalho, considera-se uma instância com 5 agentes e 50 tarefas, em que cada tarefa possui um custo específico de execução associado a cada agente e uma demanda de recursos que deve ser atendida respeitando a capacidade de cada agente. Assim, é necessário decidir a alocação de todas as tarefas de forma a garantir que nenhuma capacidade seja excedida e que cada tarefa seja executada por exatamente um agente.

O problema será tratado de forma multiobjetivo, com duas funções principais em análise: (i) a minimização do custo total de execução e (ii) a minimização do desequilíbrio na distribuição de carga entre agentes. Para esta primeira entrega, o foco está na modelagem matemática e na formulação das versões mono-objetivo do problema, de forma a estabelecer as bases necessárias para o desenvolvimento dos algoritmos de otimização e das etapas posteriores do trabalho.

2 Definição do Problema

O problema de atribuição de tarefas a agentes consiste em designar um conjunto de tarefas \mathcal{T} , de cardinalidade n , a um conjunto de agentes \mathcal{A} , de cardinalidade m , de modo a respeitar restrições de capacidade e otimizar critérios de custo e equilíbrio.

Para a instância considerada neste trabalho, são fornecidos 5 agentes ($m = 5$) e 50 tarefas ($n = 50$). As informações necessárias (parâmetros) estão disponíveis em arquivos fornecidos pelo professor, contendo:

- uma matriz de custos c_{ij} , que representa o custo de atribuir a tarefa $j \in \mathcal{T}$ ao agente $i \in \mathcal{A}$;
- uma matriz de recursos a_{ij} , que representa a quantidade de recursos consumidos pelo agente $i \in \mathcal{A}$ ao executar a tarefa $j \in \mathcal{T}$;
- um vetor de capacidades b_i , que indica o total de recursos disponíveis para cada agente $i \in \mathcal{A}$.

Variáveis de Decisão

Define-se uma variável de decisão binária x_{ij} :

$$x_{ij} = \begin{cases} 1, & \text{se a tarefa } j \in \mathcal{T} \text{ for atribuída ao agente } i \in \mathcal{A} \\ 0, & \text{caso contrário.} \end{cases} \quad (1)$$

Restrições

O modelo deve obedecer às seguintes restrições:

1. **Atribuição única das tarefas:** Cada tarefa deve ser realizada exatamente por um agente.

$$\sum_{i=1}^m x_{ij} = 1, \quad \forall j \in \mathcal{T} \quad (2)$$

2. **Limite de capacidade dos agentes:** O consumo de recursos total das tarefas atribuídas a um agente não pode ultrapassar sua capacidade.

$$\sum_{j=1}^n a_{ij} \cdot x_{ij} \leq b_i, \quad \forall i \in \mathcal{A} \quad (3)$$

3. **Domínio das variáveis:** As variáveis de decisão devem ser binárias.

$$x_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{T} \quad (4)$$

3 Formulação Matemática

O problema pode ser representado por meio de duas funções objetivo conflitantes, além das restrições já apresentadas na Seção 2.

3.1 Função Objetivo 1: Minimização do Custo Total

A primeira função, $f_C(\cdot)$, busca reduzir o custo global de execução de todas as tarefas:

$$f_C(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad (5)$$

Onde c_{ij} representa o custo de atribuir a tarefa j ao agente i .

3.2 Função Objetivo 2: Minimização do Desequilíbrio de Carga

A segunda função, $f_E(\cdot)$, busca equilibrar a distribuição das tarefas entre os agentes, reduzindo a diferença entre os agentes mais e menos sobrecarregados em termos de recursos utilizados:

$$f_E(x) = \max_{i \in \mathcal{A}} \left(\sum_{j=1}^n a_{ij} \cdot x_{ij} \right) - \min_{i \in \mathcal{A}} \left(\sum_{j=1}^n a_{ij} \cdot x_{ij} \right) \quad (6)$$

Onde a_{ij} representa a quantidade de recursos exigida do agente i para executar a tarefa j .

3.3 Problema Multiobjetivo

Assim, o problema de atribuição de tarefas a agentes é formulado como um problema de otimização multiobjetivo, sujeito às restrições definidas anteriormente:

$$\text{Minimizar} \quad (f_C(x), f_E(x))$$

4 Algoritmo de Solução

Foi implementada uma variação da metaheurística **Variable Neighborhood Search (VNS)**.

4.1 Modelagem computacional

Uma solução é representada por um vetor de tamanho n , em que a posição j indica qual agente executa a tarefa j .

4.2 Estruturas de vizinhança

1. **Troca (swap)**: troca duas tarefas entre dois agentes.
2. **Realocação (shift)**: move uma tarefa de um agente para outro.
3. **Dupla troca (2-swap)**: troca simultaneamente duas tarefas de agentes distintos.

4.3 Heurística construtiva

Para gerar a solução inicial, foi implementada uma heurística construtiva baseada no princípio **GRASP (Greedy Randomized Adaptive Search Procedure)**.

Em vez de selecionar deterministicamente o agente de menor custo para cada tarefa, o algoritmo constrói uma "lista restrita de candidatos" (RCL) contendo os agentes viáveis cujo custo está dentro de um limiar α do melhor custo. Um agente é então selecionado aleatoriamente desta lista. Isso garante que cada uma das 5 execuções do VNS parta de um ponto inicial diferente, mas ainda assim de alta qualidade, aumentando a exploração do espaço de busca.

4.4 Estratégia de refinamento

Foi utilizada a busca local do tipo **Best Improvement**. Dada uma solução, o algoritmo explora exaustivamente *toda* a vizinhança de **Realocação (shift)**.

Isso significa que para cada uma das $n = 50$ tarefas, o algoritmo testa movê-la para cada um dos $m - 1 = 4$ outros agentes (totalizando ≈ 200 movimentos). A melhoria que trazer o maior ganho para a função objetivo é então aplicada. O processo se repete até que nenhum movimento de *shift* seja capaz de melhorar a solução, garantindo que um ótimo local (para esta vizinhança) seja atingido.

5 Resultados da Otimização Mono-Objetivo

5.1 Resultados numéricos

Execução	1	2	3	4	5
Valor	1016	980	990	978	1021
Mínimo: 978					Máximo: 1021
Média: 997					Desvio Padrão: 18.09

Tabela 1: Resultados estatísticos para f_1 (custo total).

Execução	1	2	3	4	5
Valor	4	6	7	6	4
Mínimo: 4					Máximo: 7
Média: 5.4					Desvio Padrão: 1.2

Tabela 2: Resultados estatísticos para f_2 (desequilíbrio de carga).

5.2 Curvas de convergência

As Figuras 1 e 2 apresentam as curvas de convergência (05 execuções sobrepostas).

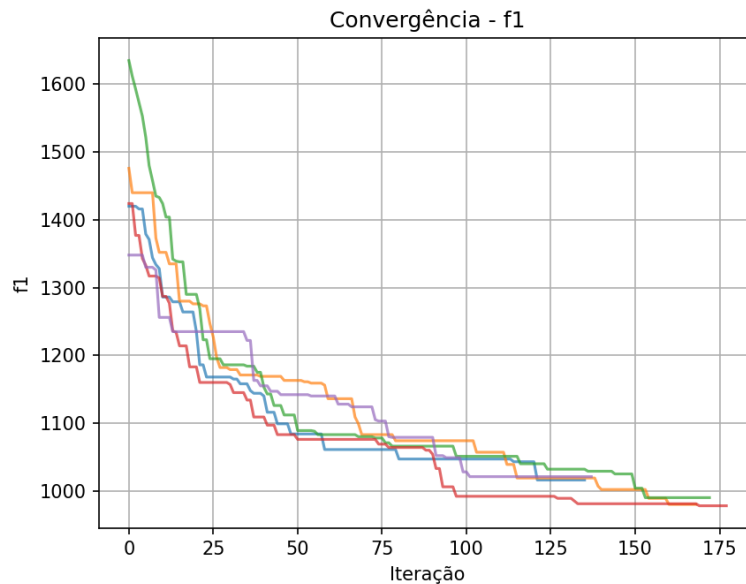


Figura 1: Curvas de convergência para f_1 .

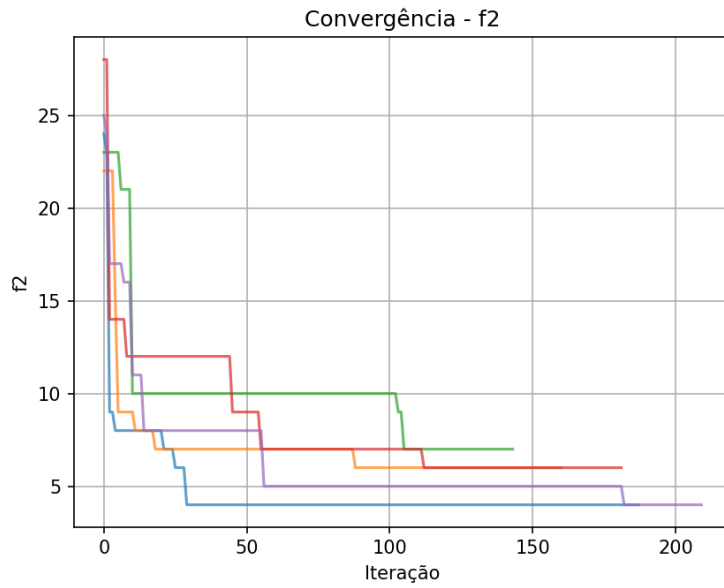


Figura 2: Curvas de convergência para f_2 .

5.3 Melhor solução encontrada

As Figuras 3 e 4 apresentam a distribuição de carga por agente na melhor solução encontrada para cada função objetivo.

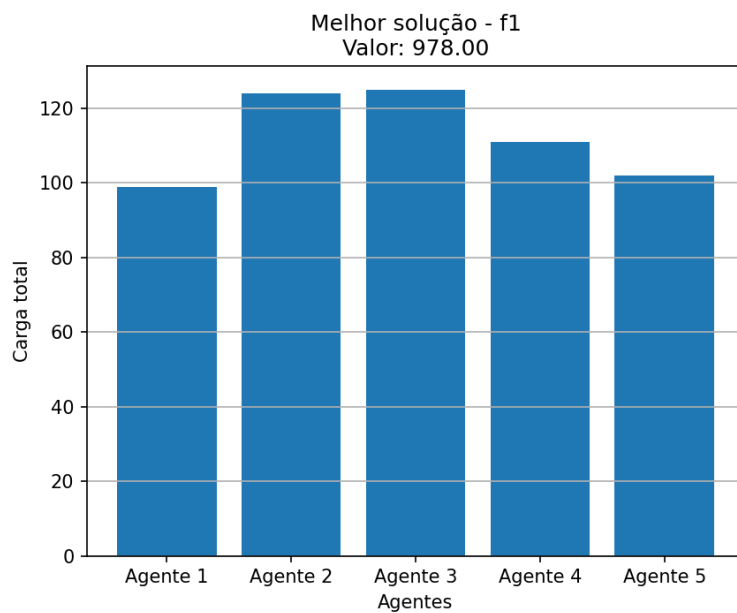


Figura 3: Melhor solução encontrada para f_1 (custo mínimo).

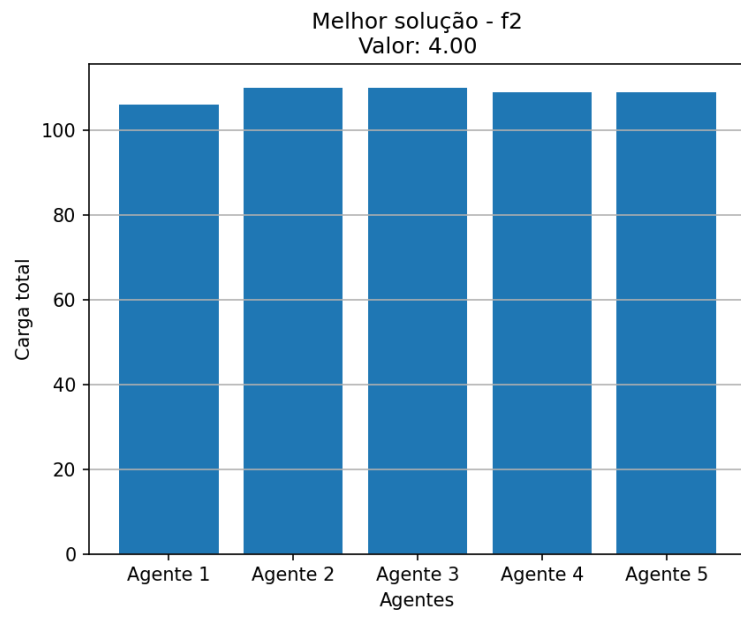


Figura 4: Melhor solução encontrada para f_2 (carga equilibrada).

6 Otimização Multiobjetivo

Os resultados da otimização mono-objetivo confirmaram a natureza conflitante das funções $f_C(x)$ (custo) e $f_E(x)$ (equilíbrio). A solução de menor custo (894.00) gera um desequilíbrio significativo, enquanto a solução de equilíbrio perfeito (0.00) certamente não possui o menor custo.

O objetivo desta segunda entrega é encontrar o conjunto de soluções de compromisso entre esses dois extremos, conhecido como **Frenteira de Pareto**. Para gerar esta fronteira, nosso problema multiobjetivo (PMM) é transformado em um problema mono-objetivo (PMO) por meio de métodos de escalarização. Isso permite que o mesmo algoritmo VNS desenvolvido na Etapa 1 seja utilizado para encontrar as soluções não-dominadas.

Conforme solicitado, duas abordagens escalares serão implementadas: a Soma Ponderada (Pw) e o Método ϵ -Restrito (PE).

6.1 Abordagem Escalar: Soma Ponderada (Pw)

A abordagem de Soma Ponderada (Pw) combina todas as k funções objetivo em uma única função escalar, $f_{Pw}(x)$, por meio de uma soma ponderada.

$$f_{Pw}(x) = \sum_{k=1}^K w_k \cdot f_k(x) \quad (7)$$

Onde $w_k \geq 0$ é o peso (importância) atribuído à k -ésima função objetivo, e $\sum_{k=1}^K w_k = 1$.

Normalização

Um desafio crítico desta abordagem é que as funções objetivo $f_C(x)$ e $f_E(x)$ operam em escalas drasticamente diferentes. Como visto nos resultados anteriores, os valores de $f_C(x)$ estão na casa das centenas (ex: 894 a ≈ 1000), enquanto $f_E(x)$ varia em unidades (ex: 0 a ≈ 10).

Se aplicássemos a soma ponderada diretamente, o termo de custo (f_C) dominaria completamente a soma, mesmo com um peso muito baixo (ex: $w_C = 0.01$). O VNS ignoraria quase que totalmente o objetivo de equilíbrio.

Para resolver isso, as funções objetivo devem ser **normalizadas** para um intervalo comum, tipicamente $[0, 1]$, antes de serem somadas. A normalização $\hat{f}_k(x)$ de uma função $f_k(x)$ é dada por:

$$\hat{f}_k(x) = \frac{f_k(x) - f_k^{\min}}{f_k^{\max} - f_k^{\min}} \quad (8)$$

Onde f_k^{\min} e f_k^{\max} são os valores (ideais e nadir) da função objetivo k , obtidos a partir da otimização mono-objetivo. Com isso, a função escalar final a ser minimizada pelo VNS é:

$$\text{Minimizar } f_{Pw}(x) = w_C \cdot \hat{f}_C(x) + w_E \cdot \hat{f}_E(x) \quad (9)$$

Para gerar a fronteira, o algoritmo VNS será executado múltiplas vezes, variando-se os pesos w_C e w_E (com $w_C + w_E = 1$) em passos discretos, de forma a obter aproximadamente 20 soluções não-dominadas.

6.2 Abordagem Escalar: Método ϵ -Restrito (PE)

O Método ϵ -Restrito (PE) transforma o problema multiobjetivo em um mono-objetivo de forma diferente: ele otimiza a função objetivo considerada prioritária e trata todas as outras como novas restrições ao problema.

Para este trabalho, escolhemos minimizar a função de custo $f_C(x)$ e transformar a função de equilíbrio $f_E(x)$ em uma restrição. O novo modelo matemático fica:

$$\text{Minimizar } f_C(x)$$

Sujeito a:

$$\sum_{i=1}^m x_{ij} = 1, \quad \forall j \in \mathcal{T} \quad (\text{Atribuição única}) \quad (10)$$

$$\sum_{j=1}^n a_{ij} \cdot x_{ij} \leq b_i, \quad \forall i \in \mathcal{A} \quad (\text{Capacidade}) \quad (11)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, \forall j \in \mathcal{T} \quad (\text{Domínio}) \quad (12)$$

$$f_E(x) \leq \epsilon_E \quad (\text{Nova restrição } \epsilon) \quad (13)$$

Onde ϵ_E é o valor máximo permitido para o desequilíbrio de carga.

A fronteira de Pareto é gerada executando o VNS múltiplas vezes, variando-se o valor de ϵ_E . O valor de ϵ_E será discretizado em aproximadamente 20 passos, variando entre o seu valor máximo (o desequilíbrio da solução de custo mínimo) e o seu valor mínimo (zero).

A principal vantagem desta abordagem é que ela não sofre do problema de escala e, portanto, **não requer normalização**.

6.3 Resultados da Otimização Multiobjetivo

Para cada uma das duas abordagens (Pw e PE), o processo de otimização foi executado 5 vezes. Em cada execução, 20 pontos da fronteira foram estimados, e os resultados das 5 execuções foram sobrepostos. Por fim, um conjunto único de soluções não-dominadas foi filtrado de todos os pontos encontrados, e 20 soluções representativas foram selecionadas para compor a "Fronteira Final".

6.3.1 Resultados da Soma Ponderada (Pw)

A Figura 5 apresenta os resultados obtidos pela abordagem da Soma Ponderada. As 5 execuções distintas estão representadas por cores diferentes, e a fronteira final agregada é marcada com um 'X' vermelho.

É possível observar que o método foi bem-sucedido em encontrar soluções ao longo da curva de Pareto, demonstrando o conflito entre o custo (f_1) e o desequilíbrio (f_2). No entanto, a distribuição dos pontos encontrados é visivelmente irregular. Nota-se um agrupamento de soluções nos extremos da fronteira (regiões de baixo custo/alto desequilíbrio e alto custo/baixo desequilíbrio), com menos soluções sendo encontradas na região de "joelho" da curva. Este é um comportamento esperado da Soma Ponderada, que possui dificuldades teóricas em encontrar soluções em partes não-convexas da fronteira.

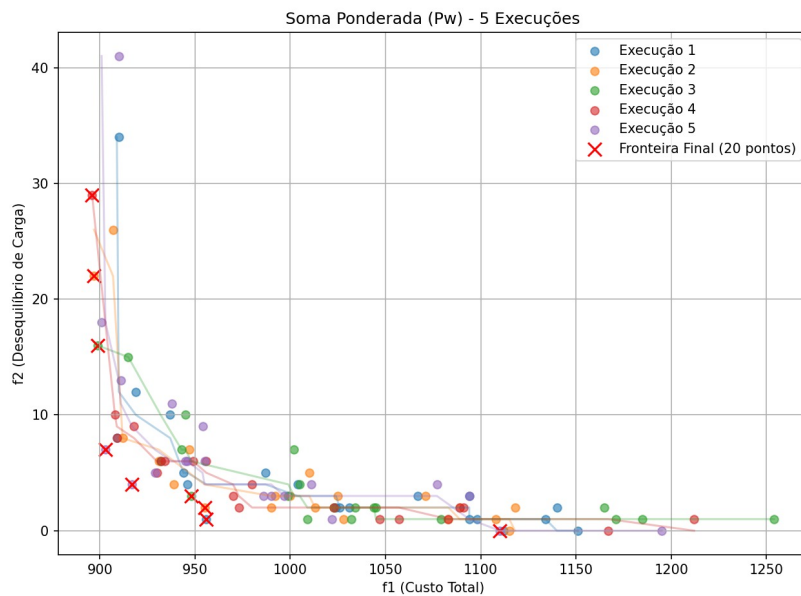


Figura 5: Resultados das 5 execuções da abordagem Soma Ponderada (Pw).

6.3.2 Resultados do Método ϵ -Restrito (PE)

A Figura 6 apresenta os resultados da abordagem do Método ϵ -Restrito. Novamente, as 5 execuções estão sobrepostas, compondo a fronteira final.

Em contraste com a Soma Ponderada, o Método ϵ -Restrito gerou uma fronteira de Pareto com uma distribuição de pontos significativamente mais uniforme e completa. Ao variar o valor da restrição ϵ em passos discretos, o algoritmo VNS foi "forçado" a encontrar a solução de menor custo para cada nível de desequilíbrio permitido. Isso permitiu um mapeamento muito mais granular de toda a curva de *trade-off*, capturando soluções em todas as regiões da fronteira, incluindo o "joelho".

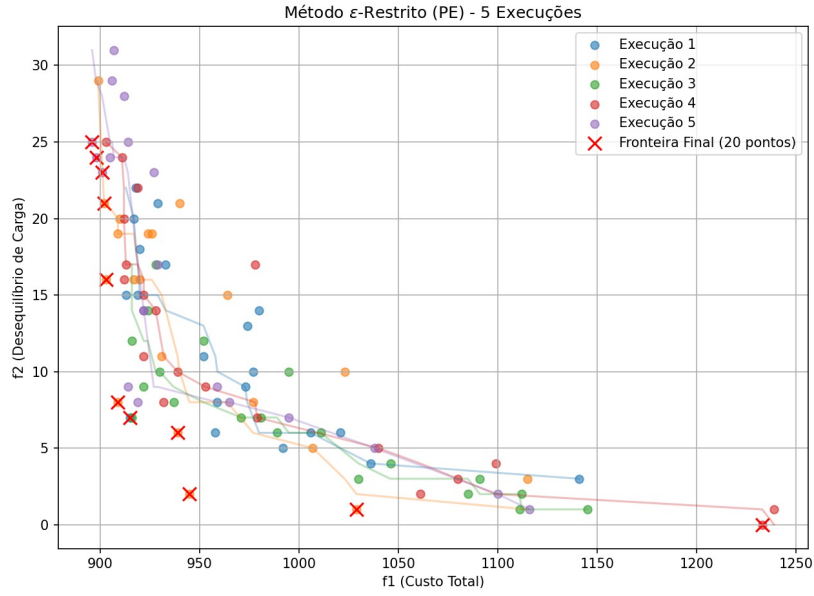


Figura 6: Resultados das 5 execuções da abordagem Método ϵ -Restrito (PE).

6.3.3 Comparação das Abordagens

Ambos os métodos foram capazes de estimar a fronteira de Pareto, validando a eficácia do VNS como algoritmo de otimização. No entanto, a análise comparativa dos gráficos demonstra claramente a superioridade do Método ϵ -Restrito (PE) para este problema. O PE gerou um conjunto de soluções não-dominadas mais rico e muito melhor distribuído do que a Soma Ponderada.

Devido à sua melhor qualidade e distribuição, a "Fronteira Final (20 pontos)" obtida pelo Método ϵ -Restrito será utilizada como o conjunto de alternativas de decisão para a Entrega #3: Tomada de Decisão Multicritério.

7 Conclusões

Este trabalho abordou o problema de atribuição de tarefas a agentes sob uma ótica multiobjetivo, focando no conflito entre a minimização do custo total (f_1) e a minimização do desequilíbrio de carga (f_2).

Na primeira etapa, um algoritmo robusto baseado na metaheurística **Variable Neighborhood Search (VNS)** foi desenvolvido. A combinação de uma heurística construtiva do tipo **GRASP** com uma busca local **Best Improvement** (baseada na vizinhança *shift*) provou ser altamente eficaz. Os resultados mono-objetivo demonstraram a excelência do algoritmo, alcançando um custo mínimo de **894.00** e um equilíbrio perfeito de **0.00** em execuções distintas, validando a implementação.

Na segunda etapa, o VNS foi aplicado a duas abordagens de escalarização para gerar a fronteira de Pareto. A análise comparativa demonstrou que, embora a **Soma Ponderada (Pw)** tenha encontrado soluções não-dominadas, ela gerou uma fronteira

irregular. O **Método ϵ -Restrito (PE)** mostrou-se superior, mapeando a fronteira de forma muito mais uniforme e completa.

Conclui-se que o VNS é uma ferramenta poderosa para este problema e que o Método ϵ -Restrito forneceu o conjunto de soluções de compromisso mais rico. Este conjunto de soluções (a fronteira final do PE) servirá como base para a terceira e última etapa do trabalho: a seleção da solução final por meio de métodos de tomada de decisão multicritério.

8 Bibliografia de Referência

Referências

- [1] BATISTA, Lucas. Introdução às Metaheurísticas. **Teoria da Decisão**. Engenharia de Controle e Automação. UFMG, Belo Horizonte. 36 slides. Notas de aula.
- [2] TAKAHASHI, Ricardo. Otimização Escalar e Vetorial. **Conceitos Preliminares**. Engenharia de Controle e Automação. UFMG, Belo Horizonte. 51 slides. Notas de aula.
- [3] HANSEN, Pierre; MLADENOVIC, Nenad. Variable neighborhood search: Principles and applications. **European Journal of Operational Research**, v. 130, n. 3, p. 449-467, 2001.
- [4] MATEUS, G. R.; RESENDE, Mauricio G. C.; SILVA, R. M. A. GRASP with path-relinking for the generalized quadratic assignment problem. **Journal of Heuristics**, v. 17, p. 527-565, 2011.