LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

META-HEURÍSTICA VNS APLICADA A UM PROBLEMA DE SEQUENCIAMENTO JUSTO

João Henrique de A. Pereira
Universidade Federal da Paraíba
João Pessoa - PB - Brasil
jhap@academico.ufpb.br

Bruno J. S. Pessoa
Universidade Federal da Paraíba
João Pessoa - PB - Brasil
bruno@ci.ufpb.br

Caroline Rocha
IVADO Labs
Montréal - Québec - Canadá
caroline.rocha@ivadolabs.com

Daniel Aloise
Polytechnique Montréal
Montréal - Québec - Canadá
daniel.aloise@polymtl.ca

RESUMO

O mRTP (do inglês, *Minmax Response Time Problem*) é um problema de sequenciamento justo que emerge sempre que tarefas, clientes, produtos ou eventos precisam ser sequenciados de modo a minimizar a variabilidade do tempo esperado entre alocações sucessivas de recursos escassos. Com aplicações que variam do escalonamento de plantões médicos à manutenção de máquinas industriais, o mRTP não tem sido tão explorado na literatura quando comparado a outros problemas de sequências justas. Este trabalho propõe estruturas de dados especializadas para o mRTP, as quais, aliadas a um algoritmo proposto baseado na meta-heurístca VNS, obteve as melhores soluções para o conjunto de instâncias referência da literatura até o momento.

PALAVRAS CHAVE. Escalonamento, Sequências Justas, Meta-heurísticas.

Área Principal: MH - Meta-heurísticas

ABSTRACT

The Minmax Response Time Problem (mRTP) is a fair sequence problem that arises whenever tasks, clients, products, or events need to be sequenced to minimize the variability of the waiting time between successive allocations of scarce resources. With applications ranging from scheduling medical shifts to maintaining industrial machinery, the mRTP has not been as extensively explored in the literature compared to other fair sequence problems. This work proposes specialized data structures for the mRTP, which, combined with a proposed algorithm based on the

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

VNS metaheuristic, have achieved the best solutions for the benchmark instance set in the literature to date.

KEYWORDS. **Scheduling. Fair Sequences. Metaheuristics.**

Main Area: MH - Metaheuristics

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

1. Introdução

Problemas de alocação de recursos escassos são constantemente observados nos mais diversos contextos. No que diz respeito à alocação de recursos, o conceito de sequências justas, proposto por Kubiak et al. [2004], define problemas de escalonamento aos quais são impostas restrições relativas às distâncias temporais entre sucessivas execuções de uma determinada tarefa, normalmente com a finalidade de reduzir a variabilidade de tais distâncias ao longo do tempo. Uma das formas de alcançar esse objetivo é minimizar as máximas distâncias entre execuções sucessivas de uma mesma tarefa. Na literatura a respeito das sequências justas, destacam-se o Problema da Variabilidade do Tempo de Resposta (RTVP, do inglês *Response Time Variability Problem*) e o mRTP. O RTVP é um problema de otimização NP-Difícil, que surge sempre que tarefas, clientes, produtos ou eventos precisam ser sequenciados de modo a minimizar a variabilidade do tempo esperado entre alocações sucessivas de recursos escassos [Corominas et al., 2007]. O mRTP [Salhi e García-Villoria, 2012], que pode ser considerado como uma versão minmax do RTVP, é o problema de sequências justas abordado neste trabalho.

O mRTP possui aplicações práticas que permeiam diversos setores da sociedade. No setor da saúde, por exemplo, o escalonamento de plantões médicos em Unidades de Terapia Intensiva (UTIs) é um dessas aplicações [Erhard et al., 2018]. Além da dificuldade usual de elaboração de uma escala de trabalho para um serviço de natureza ininterrupta, a complexidade na elaboração da escala de plantões em UTIs é acentuada por alguns fatores que são específicos ao profissional da área da saúde. Por exemplo, caso os médicos optem por concentrar todos os seus plantões em uma única semana, plantões consecutivos podem ficar muito próximos, a ponto de comprometer o desempenho dos profissionais e também violar a legislação trabalhista. Por outro lado, se o intervalo temporal entre plantões consecutivos for muito longo, a continuidade no tratamento será prejudicada, já que as condições clínicas dos pacientes em UTIs mudam muito rapidamente. O fato de que o número de plantões mensais de cada médico pode variar conforme contratos individuais agrava ainda mais a situação. Uma solução para esse problema é escalonar os plantões de modo que os intervalos de tempo entre plantões consecutivos de um mesmo médico sejam o mais uniformes possível.

O mRTP também pode ser aplicado ao planejamento de manutenções periódicas de máquinas [Anily et al., 1998; Bar-Noy et al., 2002]. Nesse contexto, empresas são contratadas para realizar serviços de manutenção e estabelecem os valores do contrato de acordo com o número de visitas realizadas em um determinado período. Assim como ocorre no exemplo anterior, busca-se evitar duas situações: a primeira é a realização de manutenções desnecessárias, que ocorre quando os serviços são agendados muito próximos uns dos outros; a segunda, por sua vez, é o aumento dos custos dos serviços de manutenção, que tendem a crescer à medida que o tempo entre uma manutenção e outra se alonga. Portanto, o objetivo é elaborar um cronograma de serviços de manutenção que minimize a variabilidade do tempo entre manutenções consecutivas.

O escalonamento de processos em sistemas de tempo real também está contemplado pelo mRTP [Waldspurger e Weihl. W., 1995]. Em certos sistemas de tempo real, é desejável implementar um escalonamento de processos utilizando o controle de compartilhamento proporcional, com o objetivo de minimizar o tempo de resposta de cada processo quando solicitado. Para alcançar esse objetivo, os processos precisam ser executados a uma taxa constante ao longo do tempo. Em outras palavras, os intervalos de tempo entre execuções consecutivas devem ser tão uniformes quanto possível.

Introduzido por Corominas et al. [2007], o RTVP foi amplamente estudado na última década. Diversas soluções foram propostas, abrangendo tanto métodos exatos quanto heurísticos. Corominas et al. [2010] propuseram um modelo baseado em Programação Linear Inteira Mista com vários melhoramentos com relação ao modelo original, aumentando de forma significativa o

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

número de instâncias resolvidas até a otimalidade. García-Villoria et al. [2013], por sua vez, desenvolveram um método *branch and bound* (B&B) específico, voltado para as particularidades do RTVP. A solução proposta obteve os melhores resultados até então, em termos de desempenho, quando comparada com os outros métodos exatos da literatura. Já com relação a métodos heurísticos, destacam-se as meta-heurísticas *Variable Neighborhood Search* (VNS), *Simulated Annealing* e Algoritmos Genéticos, descritas nos trabalhos de Corominas et al. [2011], García-Villoria e Pastor [2013b] e García-Villoria e Pastor [2010], respectivamente.

Apresentado originalmente em [Salhi e García-Villoria, 2012], o mRTP não teve ao longo do tempo a mesma atenção do RTVP. Os autores do trabalho inicial propuseram uma heurística gulosa para o problema, originalmente concebida para resolver o RTVP, contudo sem apresentar nenhum indicador de qualidade das soluções obtidas. Para preencher tal lacuna, García-Villoria e Pastor [2013] propuseram nove procedimentos de busca local, além de cinco algoritmos baseados em meta-heurísticas para resolvê-lo, provendo vários *benchmarks* para o mRTP.

Inspirado no trabalho de Rocha et al. [2022], que propõe um solução heurística para o Problema das Sequências Justas ponderadas, proposto por Pessoa et al. [2018], o presente trabalho propõe estruturas de dados especializadas para o mRTP, além de um framework baseado na meta-heurística VNS para resolvê-lo. Experimentos computacionais constatam que a meta-heurística proposta obteve as melhores soluções para o conjunto de instâncias referência da literatura até o momento.

O restante do artigo é organizado como segue: A Seção 2 apresenta uma definição formal do problema. A Seção 3 descreve as estruturadas de dados especializadas propostas. A Seção 4 apresenta a solução geral proposta, baseada na meta-heurística VNS. Por fim, a Seção 5 discute os resultados dos experimentos computacionais, e a Seção 6 contempla as considerações finais.

2. Definição formal do problema

Formalmente, o problema trata do sequenciamento de símbolos $i \in \{1, 2, ..., n\}$, onde cada símbolo possui d_i cópias. Sejam $D = \sum_{i=1}^{n} d_i$ o tamanho das sequências e $\bar{t_i} = D/d_i$ distância ideal entre duas cópias consecutivas de cada símbolo i. Uma solução viável S é uma sequência circular contendo um total de D cópias tal que $S = (s_1, s_2, ...s_D)$, onde s_h é a cópia posicionada na posição h da sequência. Para cada símbolo i com $d_i \geq 2$, t_k^i é a distância entre as cópias k e $k+1$ do símbolo i. A distância entre duas cópias em posições consecutivas h e $h+1$ é considerada igual a 1 e, como a sequência é circular, a posição 1 vem logo após a posição D da sequência. Além disso, a distância entre a última cópia do símbolo i de um ciclo e sua primeira cópia do próximo ciclo é $t_{d_i}^i$. Por fim, para os símbolos i onde $d_i = 1$, $t_k^i = \bar{t_i} = D$. O objetivo do mRTP é minimizar o erro absoluto máximo entre as distâncias t_k^i de cada cópia e a distância ideal $\bar{t_i}$. Dessa forma, podemos escrever a função objetivo a ser minimizada como:

$$maxRT = \max_{i=1}^{n}\{\max_{k=1}^{d_i}\{|t_k^i - \bar{t_i}|\}\}. \quad (1)$$

A Figura 1 ilustra uma solução viável para mRTP, destacando o caráter cíclico da sequência de símbolos. Para os símbolos $i \in \{1, 2, 3, 4\}$, temos que $d_1 = 3$, $d_2 = 2$, $d_3 = 2$, $d_4 = 1$. Dado que $D = 8$, as distâncias ideais para cada símbolo são $\bar{t_1} = \frac{8}{3}$, $\bar{t_2} = 4$, $\bar{t_3} = 4$ e $\bar{t_4} = 8$. Quanto às distâncias entre as cópias, tomando como exemplo o símbolo 1, tais valores são $t_1^1 = 4$, $t_2^1 = 2$, $t_3^1 = 2$. O erro absoluto máximo para essa solução pode então ser calculado da seguinte forma, de acordo com (1):

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

$$\begin{aligned} maxRT = \max\{&\max\{|4-8/3|, |2-8/3|, |2-8/3|\},\\ &\max\{|3-4|, |5-4|\},\\ &\max\{|2-4|, |6-4|\},\\ &\max\{|8-8|)\}\} = 2. \end{aligned}$$

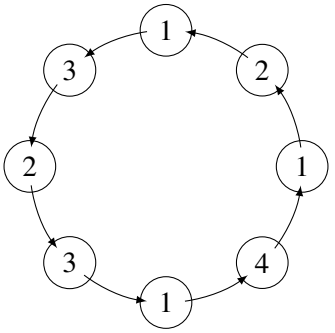


Figura 1: Exemplo de solução viável para o mRTP.

3. Estruturas de dados especializadas

A estrutura de dados utilizada é inspirada na estrutura proposta em Rocha et al. [2022] para resolver o Problema das Sequências Justas ponderadas proposto por Pessoa et al. [2018]. Por meio dela, podemos avaliar se um movimento é capaz de melhorar a solução corrente em tempo constante. Para isso, uma solução S do mRTP é tratada como uma lista unidimensional de D cópias, com cada cópia contendo seguintes informações:

- $\Gamma^+[i,h]$: posição da primeira cópia de i que aparece após a posição h;
- $\Gamma^-[i,h]$: posição da primeira cópia de i que aparece antes da posição h;
- $d(h)$: distância entre a cópia do símbolo s_h na posição h e a próxima cópia de s_h em S, ou seja:

$$d(h) = \begin{cases} \Gamma^+[s_h,h] - h, \text{ se } \Gamma^+[s_h,h] \geq h \\ \Gamma^+[s_h,h] - h + D, \text{ caso contrário.} \end{cases}$$

- $\sigma(h)$: valor da função objetivo associado à cópia h de S:

$$\sigma(h) = |d(h) - \bar{t}_{s_h}|.$$

A Figura 2 ilustra o conteúdo dessa estrutura de dados para a sequência $S = 1,3,2,3,1,4,1,2$. Vale destacar que o valor máximo de $\sigma(h)$ também é mantido na estrutura de dados, permitindo, em tempo constante, avaliar se há melhora nos movimentos utilizados. Também é importante mencionar que as estruturas não armazenam as listas Γ^+ e Γ^- dos símbolos que possuem d_i = 1, tendo em vista que seu impacto é desprezível nos cálculos.

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

S	1	3	2	3	1	4	1	2
h	1	2	3	4	5	6	7	8
$d(h)$	4	2	5	6	2	8	2	3
$\sigma(h)$	$\frac{4}{3}$	2	1	2	$\frac{2}{3}$	0	$\frac{2}{3}$	1
$\Gamma^+[1,h]$	5	5	5	5	7	7	1	1
$\Gamma^+[2,h]$	3	3	8	8	8	8	8	3
$\Gamma^+[3,h]$	2	4	4	2	2	2	2	2
$\Gamma^-[1,h]$	7	1	1	1	1	5	5	7
$\Gamma^-[2,h]$	8	8	8	3	3	3	3	3
$\Gamma^-[3,h]$	4	4	2	2	4	4	4	4

Figura 2: Exemplo de aplicação da estrutura de dados utilizada.

4. Estrutura da Meta-heurística

Nesta seção, serão detalhados os procedimentos que compõem a solução heurística proposta neste trabalho. Ela é baseada na estrutura do VNS (*Variable Neighborhood Search*) proposta em Hansen et al. [2017] e possui como ideia geral partir de uma solução inicial obtida por meio de um algoritmo randomizado e executar, ao longo das iterações, uma busca local determinada pelo VND (*Variable Neighborhood Descent)* em uma vizinhança da melhor solução corrente obtida pelo procedimento de perturbação, que consiste na mudança sistemática de vizinhanças utilizada para explorar o espaço de soluções.

4.1. Solução inicial

A solução inicial é obtida por meio do algoritmo randomizado de escalonamento lotérico proposto por Waldspurger e Weihl [1994]. O algoritmo adiciona uma cópia por vez na solução, até que todas as cópias tenham sido alocadas. Para cada posição, o símbolo a ser alocado é escolhido aleatoriamente e, ao longo das iterações, a probabilidade de um símbolo ser alocado é calculada dividindo o número de cópias remanescentes do símbolo pelo número total de cópias de todos os símbolos a serem inseridos na solução.

4.2. Busca Local

As seguintes estruturas de vizinhança são utilizadas na Busca Local:

- $ShiftOne$ - Duas cópias que estão em posições consecutivas trocam de posição;
- $Swap$ - Duas cópias em quaisquer posições da solução trocam de posição.

A Busca Local utiliza essas vizinhanças no esquema de *Variable Neighborhood Descent* (VND). Dada uma solução corrente S, o procedimento tenta encontrar uma primeira melhora na vizinhança $ShiftOne$. Caso não encontre, o algoritmo tenta encontrar uma primeira melhora na vizinhança $Swap$. Se uma melhoria é encontrada, o procedimento é repetido, do contrário, um ótimo local com relação a ambas estruturas de vizinhança foi encontrado. A avaliação de um movimento por meio dessas vizinhanças é feita em $O(1)$, enquanto a atualização de uma solução é feita em $O(n \times D)$.

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

Com o objetivo de explorar regiões maiores do espaço de solução, a busca utiliza o critério de aceitação de melhoria proposto em García-Villoria e Pastor [2013a]. Assim, uma solução S é considerada melhor do que a outra solução S' se:

i. $cost(S) < cost(S')$, isto é, o custo de S for menor do que o custo de S' ;

ii. Em caso de empate, se $\#cost(S) < \#cost(S')$, onde $\#cost(Sol)$ é o número de intervalos dominantes de uma solução. Ou seja, o número de posições h para as quais $\sigma(h)$ é igual a $cost(Sol)$;

iii. Caso o empate persista, se $RTV(S) < RTV(S')$, onde $RTV(Sol)$ é a função objetivo do Problema da Variabilidade do Tempo de Resposta proposto em [Corominas et al., 2007], que pode ser calculada como: $RTV(Sol) = \sum_{h=1}^{D} \sigma(h)^2$.

A ordem das vizinhanças foi escolhida considerando a complexidade de exploração associada a cada vizinhança, tendo em vista que, enquanto $ShiftOne$ esgota as soluções vizinhas possíveis em $O(D)$, a vizinhança $Swap$ possui complexidade de exploração em $O(D^2)$. Além disso, foi observada uma melhor performance do algoritmo utilizando a estratégia de *firstImprovement*, em ambas as vizinhanças. Assim, a estrutura proposta para o VND é:

Algoritmo 1: VND

Procedimento: VND(S)
1 $S' \leftarrow S$;
2 **repeat**
3 $\quad S \leftarrow S'$;
4 $\quad S' \leftarrow$ *FirstImprovingShiftOne*(S');
5 $\quad S' \leftarrow$ *FirstImprovingSwap*(S');
6 **until** S' *não for melhor que* S;
7 **return** S;

4.3. Perturbação

Na estrutura do VNS, o procedimento de pertubação serve para que a busca consiga escapar de soluções que representam mínimos locais. No algoritmo proposto, utilizamos 3 estruturas de vizinhança com esse objetivo. São elas:

- Redistribuição lotérica(N_1): Retira da solução uma porcentagem aleatória das cópias, limitada por uma constante α, e as redistribui utilizando a mesma estratégia do algoritmo randomizado que gera a solução inicial;
- Shift(N_2): Escolhe aleatoriamente uma cópia em uma posição h e a reinsere em uma posição da solução;
- 2-opt(N_3): Escolhe aleatoriamente um conjunto de cópias que aparecem consecutivamente na solução e as reinsere de forma invertida.

Por fim, podemos ver a estrutura do esquema VNS proposto no Algoritmo 2. De início, a solução inicial é obtida pelo *lotteryScheduling* (linha 3). Após isso, será criado um vizinho da melhor solução encontrada até então (linha 8) e será realizada uma busca local determinada pelo VND (linha 9). É importante salientar que, caso a vizinhança escolhida seja N_1, um parâmetro α que

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

Algoritmo 2: Variable Neighbourhood Search (VNS)

```
   Procedimento : VNS(iter_{max,t_max})
1  iter ← 1;
2  k_max ← 3;
3  best ← lotteryScheduling(D, d);
4  while iter ≤ iter_max do
5      current ← best;
6      k ← 1;
7      while k ≤ k_max do
8          current ← N_k(best);
9          current ← VND(current);
10         if cost(current) < cost(best) then
11             best ← current;
12             k ← 1;
13             iter ← 0;
14         else
15             k ← k + 1;
16     iter ← iter + 1;
17 return best;
```

limita a proporção de cópias a serem redistribuídas será considerado. Caso a busca local encontre uma melhoria (linha 10), a melhor solução é atualizada e os critérios de parada são reiniciados com o objetivo de explorar o máximo possível o espaço de soluções (linhas 11-13). Caso contrário, o algoritmo troca a vizinhança da melhor solução (linha 15) e reinicia a busca. O algoritmo é encerrado quando o número máximo de iterações ($iter_{max}$) é atingido.

5. Resultados computacionais

Nesta seção, são apresentados os resultados referentes à meta-heurística VNS proposta e é realizada uma comparação com os resultados obtidos pelas heurísticas propostas em García-Villoria e Pastor [2013a]. Os experimentos foram realizados em uma máquina com processador 12th Gen Intel Core i7-12700 utilizando o sistema operacional Ubuntu 22.04 LTS, e o algoritmo proposto foi codificado utilizando C++.

O conjunto de instâncias utilizado é o mesmo proposto para o RTVP, disponível em `https://www.ioc.upc.edu/EOLI/research/`. Essas instâncias são agrupadas em 4 categorias que são definidas pelo tamanho das sequências, cada uma contendo 15 instâncias de treino e 185 instâncias de teste. A Tabela 1 explica as distribuições de D e n de cada categoria utilizada.

	CAT1	**CAT2**	**CAT3**	**CAT4**
D	U(25, 50)	U(50, 100)	U(100, 200)	U(200, 500)
n	U(3, 15)	U(3, 30)	U(3, 65)	U(3, 150)

Tabela 1: Distribuição uniforme de D e n.

Com relação à parametrização do VNS proposto, o valor de $iter_{max}$ utilizado foi $300 \times D$, e o valor de α, que corresponde à porcentagem máxima de cópias a serem realocadas na perturbação N_1, foi fixado em 20. Ambos os parâmetros foram escolhidos de forma empírica a partir de testes preliminares realizados nas instâncias.

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

Para comparar o método proposto com o estado da arte do problema, foram consideradas as 185 instâncias de teste de cada categoria, 740 no total. Nos experimentos realizados por García-Villoria e Pastor [2013a], várias meta-heurísticas tiveram seus desempenhos comparados, dentre as quais se destacaram o VNS e o *Simulated Anneling* propostos pelos autores, ambas executadas com o critério de parada sendo atingir o tempo limite de 3600 segundos. Na Tabela 2, a coluna **GVP** representa a média dos valores das soluções, obtidos pela melhor meta-heurística de García-Villoria e Pastor [2013a] por categoria e, na coluna **Tempo-GVP** o tempo de execução de cada instância, enquanto a coluna **VNS**, por sua vez, apresenta os resultados obtidos pelo método VNS proposto, também agrupados por categoria. Por fim, os tempos médios de execução do VNS são expostos na coluna **Tempo**.

Categoria	GVP	Tempo-GVP(s)	VNS	Tempo(s)	Gap(%)
CAT1	1,060	3600	1,060	1,17	0,00
CAT2	1,107	3600	**1,101**	29,32	-0,54
CAT3	1,231	3600	**1,169**	182,38	-5,04
CAT4	1,538	3600	**1,403**	556,72	-8,78
Média	1,237	3600	**1,183**	192,40	-4,37

Tabela 2: Comparação entre os melhores resultados obtidos em [García-Villoria e Pastor, 2013a] e o VNS proposto.

Como pode ser observado na Tabela 2, os resultados do VNS proposto se equiparam (no caso das instância da categoria CAT1) ou superam os resultados das melhores meta-heurísticas já implementadas para o mRTP em termos de valor objetivo das soluções. Além disso, é notável o ganho expressivo na performance do algoritmo proposto em comparação com os descritos anteriormente, resolvendo as instâncias em uma fração do tempo necessário nas meta-heurísticas estado da arte.

Além da comparação com o estado da arte, para avaliar a efetividade da meta-heurística, foram comparados os resultados obtidos com os limites inferiores existentes e os resultados exatos. Na Tabela 3, a coluna **LB** contém os limitantes inferiores obtidos para cada categoria, enquanto a coluna **OPT** representa as soluções exatas existentes para cada categoria. García-Villoria e Pastor [2013a] foram capazes de obter soluções exatas apenas para as instâncias de teste da primeira categoria, nas demais categorias obtiveram apenas os limitantes inferiores.

Categoria	LB	VNS	OPT
CAT1	0,794	1,060	1,060
CAT2	0,842	1,101	*
CAT3	0,887	1,169	*
CAT4	0,917	1,403	*

Tabela 3: Comparação entre dos resultados obtidos e os limitantes inferiores e resultados exatos obtidos em [García-Villoria e Pastor, 2013a].

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

A Tabela 3 mostra que o algoritmo proposto foi capaz de encontrar as soluções ótimas na CAT1 e que obteve sucesso na aproximação com os limites inferiores nas demais categorias. No entanto, é válido destacar que, como pode ser observado pela diferença entre os valores das colunas **LB** e **OPT** para a CAT1, a comparação com os limitantes inferiores não é uma métrica precisa para medir a distância entre os resultados do algoritmo proposto e a solução ótima.

6. Considerações finais

O presente trabalho abordou um problema escalonamento, denominado de mRTP, que consiste na versão minmax do Problema da Variabilidade do Tempo de Resposta (RTVP, do inglês *Response Time Variability Problem*). O mRTP possui uma ampla gama de aplicações práticas que variam desde o escalonamento de plantões médicos em UTIs à manutenções periódicas de máquinas industriais. Como solução para o problema, propusemos estruturas de dados especializadas que foram responsáveis por acelerar de forma considerável a busca local do algoritmo proposto. Aliadas a um framework baseado na meta-heurística VNS, bem como em uma combinação nova de estruturas de vizinhança, o algoritmo proposto encontrou as melhores soluções para as instâncias referência para o mRTP até o momento. Como trabalho futuro, consideramos estender a solução geral proposta de maneira a englobar todas as variações do RTVP.

Referências

Anily, S., Glass, C. A., e Hassin, R. (1998). The scheduling of maintenance service. *Discrete Applied Mathematics*, 82(1):27–42. ISSN 0166-218X.

Bar-Noy, A., Bhatia, R., Naor, J. S., e Schieber, B. (2002). Minimizing service and operation costs of periodic scheduling. *Math. Oper. Res.*, 27(3):518–544. ISSN 0364-765X.

Corominas, A., García-Villoria, A., e Pastor, R. (2011). Metaheuristic algorithms hybridised with variable neighbourhood search for solving the response time variability problem. *Top*, 21:1–17.

Corominas, A., Kubiak, W., e Palli, N. M. (2007). Response time variability. *Journal of Scheduling*, 10(2):97–110. ISSN 1099-1425.

Corominas, A., Kubiak, W., e Pastor, R. (2010). Mathematical programming modeling of the response time variability problem. *European Journal of Operational Research*, 200(2):347–357. ISSN 0377-2217.

Erhard, M., Schoenfelder, J., Fügener, A., e Brunner, J. O. (2018). State of the art in physician scheduling. *European Journal of Operational Research*, 265(1):1 – 18. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S0377221717305787.

García-Villoria, A., Corominas, A., Delorme, X., Dolgui, A., Kubiak, W., e Pastor, R. (2013). A branch and bound algorithm for the response time variability problem. *Journal of Scheduling*, 16 (2):243–252.

García-Villoria, A. e Pastor, R. (2013). Minimising maximum response time. *Computers & Operations Research*, 40(10):2314–2321. ISSN 0305-0548.

García-Villoria, A. e Pastor, R. (2010). Solving the response time variability problem by means of a genetic algorithm. *European Journal of Operational Research*, 202:320–327.

García-Villoria, A. e Pastor, R. (2013a). Minimising maximum response time. *Computers & Operations Research*, 40(10):2314–2321. ISSN 0305-0548. URL http://dx.doi.org/10.1016/j.cor.2013.03.014.

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br

García-Villoria, A. e Pastor, R. (2013b). Simulated annealing for improving the solution of the response time variability problem. *International Journal of Production Research*, 51(16):4911–4920.

Hansen, P., Mladenović, N., Todosijević, R., e Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, 5(3):423–454. ISSN 2192-4406. URL https://www.sciencedirect.com/science/article/pii/S2192440621000873.

Kubiak, Wieslaw, Leung, e Y-T, J. (2004). *Fair sequence, em Handbook of scheduling : algorithms, models, and performance analysis*. Chapman and Hall/CRC, Boca Raton, Fla. ISBN 1584883979 9781584883975.

Pessoa, B. J. S., Aloise, D., e Cabral, L. A. (2018). The weighted fair sequences problem. *Computers & Operations Research*, 91:121–131. ISSN 0305-0548. URL https://www.sciencedirect.com/science/article/pii/S0305054817302848.

Rocha, C., de Sousa Pessoa, B. J., Aloise, D., e dos Anjos Formiga Cabral, L. (2022). An efficient implementation of a vns heuristic for the weighted fair sequences problem. Les Cahiers du GERAD G-2022-05, Groupe d'études et de recherche en analyse des décisions, GERAD, Montréal QC H3T 2A7, Canada. URL https://www.gerad.ca/en/papers/G-2022-05.

Salhi, S. e García-Villoria, A. (2012). An adaptive search for the response time variability problem. *Journal of the Operational Research Society*, 63(5):597–605. URL https://doi.org/10.1057/jors.2011.46.

Waldspurger, C. A. e Weihl. W., E. (1995). Stride scheduling: Deterministic proportional- share resource management. Technical report, Cambridge, MA, USA.

Waldspurger, C. A. e Weihl, W. E. (1994). Lottery scheduling: Flexible Proportional-Share resource management. In *First Symposium on Operating Systems Design and Implementation (OSDI 94)*, Monterey, CA. USENIX Association. URL https://www.usenix.org/conference/osdi-94/lottery-scheduling-flexible-proportional-share-resource-management.

DOI: 10.59254/sbpo-2024-193524
https://proceedings.science/p/193524?lang=pt-br