

OTIMIZAÇÃO DE PROGRAMAÇÃO DE TAREFAS DE MANUTENÇÃO

João Luis Reis e Silva, MSc.

RESUMO

Este trabalho tem como objetivo apresentar um método para solução do problema de otimização das semanas de partida para um grupo de tarefas de manutenção. A solução exata para esta classe de problema reside em aplicar métodos de programação linear inteira binária (PLIB), tais soluções apresentaram grande eficiência computacional em comparação com soluções não exatas presentes nos principais pacotes comerciais de sistemas ERP, do inglês “*enterprise resource planning*”. O objetivo do uso de PLIB visa a obtenção de uma solução exata com alta eficiência e baixo custo de modelagem computacional.

1. INTRODUÇÃO

A programação de tarefas de manutenção é uma atividade extremamente estratégica dentro de qualquer setor de Planejamento e Controle da Manutenção (PCM). Uma programação de tarefas otimizada irá garantir o cumprimento das tarefas de manutenção sem ocasionar o aparecimento ou aumento de *backlogs* e, conseqüentemente, a redução de falhas e manutenções corretivas.

O processo para determinar as datas de partidas das tarefas de manutenção comumente adotado nas indústrias quase sempre é feito pela experiência dos programadores ou por um método de tentativa e erro. Algumas vezes são realizados por ferramentas comerciais onde métodos heurísticos simplificados são utilizados. Estas metodologias empregadas quase sempre determinam datas de partidas que estão distantes do ponto ótimo global.

Neste trabalho, um modelo matemático PLIB, aplicado para o processo de otimização da programação de tarefas de manutenção, é operacionalizado através das ferramentas MatLab e Gurobi. Um grupo de 142 tarefas provenientes de uma indústria do setor de bebidas e alimentos é utilizado para validar o modelo proposto. As soluções exatas encontradas, aliadas com a alta eficiência e baixo custo da modelagem computacional, permite a aplicação deste método em qualquer tipo de atividade de programação de tarefas.

2. TAREFAS DE MANUTENÇÃO

As tarefas de manutenção são basicamente ordens de manutenção provenientes de estratégias designadas para manter um ativo sob condições operacionais e confiabilidade no cumprimento de suas funções. O termo manutenção é derivado da palavra do Latim *manus tenere*, que significa *manter o que se tem* [12].

Os tipos de estratégias de manutenção influenciam a forma como as tarefas são executadas. Esta influência é relativa ao tipo de ação e intervalo na qual devem ser executadas. Os principais tipos de estratégias de manutenção são apresentados a seguir:

- manutenção corretiva: representam as estratégias de manutenção que ocorrem após as falhas, e que visam estabelecer as funções de operação do ativo. Tais estratégias não são programadas visto que o tempo até a falha não pode ser determinado [15];
- manutenção preventiva de reparo: são estratégias de manutenção baseada no tempo, i.e., ações que são executadas em função de um tempo predeterminado. Esta estratégia executa uma ação para reparar um estado de degradação do ativo. Tais ações não implicam em substituição de um componente mas uma ação para reduzir os efeitos da degradação [5]. Exemplos de manutenção preventiva de reparo são as ações de lubrificação, limpeza e reapertos;
- manutenção preventiva de troca: similar a estratégia de preventiva de reparo, a manutenção preventiva de troca são ações baseadas no tempo onde uma ação de troca é utilizada em um dado componente do ativo para restaurar um determinado estado de degradação [13]. Exemplos de preventiva de troca são as ações de substituição de elementos filtrantes e troca de óleo;
- manutenção preditiva: são estratégias de manutenção baseada na condição do ativo, i.e., ações que são executadas em função de uma dada condição de degradação monitorada. Tal estratégia executa ações tanto de reparo quanto de troca [27]. Exemplos de manutenção preditiva são os monitoramento por ultrassom, termografia, análise de vibração dentre outras;
- manutenção detectiva: manutenções detectivas são ações para reparar modos de falha ocultos, i.e., modos de falha não detectáveis em situações normais de operação. Equipamentos que exibem este tipo de modo de falha são geralmente os equipamentos reservas ou de proteção [22];
- manutenção de melhorias: são ações de melhorias ou reprojatos que visam aumentar a manutenabilidade ou confiabilidade intrínseca de um ativo;
- manutenção proativa: a manutenção proativa utiliza técnicas de manutenção preventiva e preditiva em conjunto com análise de falha (RCA) para detectar e eliminar potenciais modos de falha [7].

A. Programação de Tarefas

Em indústrias onde o número de tarefas de manutenção é relativamente alto, i.e., acima de 300, a tarefa de balancear as semanas de partidas das tarefas de forma a não sobrecarregar as equipes executantes em determinadas semanas nem sempre é fácil e consome bastante tempo, ocasionando datas de partidas sendo definidas de maneira não otimizada.

As figuras 1 e 2 ilustram o processo de otimização de programação de tarefas de manutenção em um exemplo bastante simplificado com 12 tarefas direcionadas para uma equipe de execução com 4 mantenedores que trabalham 44 horas semanais. Na figura 1 todas as datas de partidas dos planos estão direcionadas para a primeira semana, ocasionando déficit de recursos nas semanas 1 e 3.

Utilizando os algoritmos de otimização PLIB, é possível otimizar as datas de partida do exemplo em questão fazendo com que os déficits sejam reduzidos ou até mesmo eliminados. Na figura 2, a simples mudança das datas de partidas das tarefas 4, 8, 11 e 12 possibilitou a eliminação dos déficits de Homem-hora (Hh) da equipe executante. Esta otimização não altera a periodicidade das tarefas, somente a data de partida das mesmas.

Demanda (Hh)	Período	Semana Partida	Mapa de Programação (52 semanas)					
			S1	S2	S3	S4	...	S52
12	semanal	S1	1	1	1	1
16	quinzenal	S1	1	0	1	0
24	quinzenal	S1	1	0	1	0
18	quinzenal	S1	1	0	1	0
16	semanal	S1	1	1	1	1
24	semanal	S1	1	1	1	1
12	mensal	S1	1	0	0	0
6	mensal	S1	1	0	0	0
12	quinzenal	S1	1	0	1	0
24	semanal	S1	1	1	1	1
24	quinzenal	S1	1	0	1	0
24	mensal	S1	1	0	0	0
Capacidade Líquida =			160	160	160	160
Demanda Programada =			212	76	170	76
Resíduo =			-52	84	-10	84

Fig. 1 - Exemplo de mapa de programação de tarefas não otimizado.

Demanda (Hh)	Período	Semana Partida	Mapa de Programação (52 semanas)					
			S1	S2	S3	S4	...	S52
12	semanal	S1	1	1	1	1
16	quinzenal	S1	1	0	1	0
24	quinzenal	S1	1	0	1	0
18	quinzenal	S2	0	1	0	1
16	semanal	S1	1	1	1	1
24	semanal	S1	1	1	1	1
12	mensal	S1	1	0	0	0
6	mensal	S2	0	1	0	0
12	quinzenal	S1	1	0	1	0
24	semanal	S1	1	1	1	1
24	quinzenal	S2	0	1	0	1
24	mensal	S2	0	1	0	0
Capacidade Líquida =			160	160	160	160
Demanda Programada =			140	148	128	118
Resíduo =			20	12	32	42

Fig. 2 - Exemplo de mapa de programação de tarefas otimizado.

3. MODELO MATEMÁTICO

A formulação matemática adotada para o problema de otimização de tarefas de manutenção foi baseado na figura 3. Esta figura representa uma generalização para os exemplos apresentados pelas figuras 1 e 2.

A variável x_{tw} , conforme definido em (1),

$$x_{tw} = \begin{cases} 1, & \text{Existe programação para a tarefa } t \\ & \text{na semana } w \\ 0, & \text{Não existe programação} \end{cases} \quad (1)$$

representa a decisão do modelo PLIB em utilizar uma determinada tarefa t alocada na semana w no processo de atendimento das restrições impostas pelo modelo. A função objetivo, restrições e domínio das variáveis do modelo matemático, são dados por:

$$x^* = \min_x f(x) \quad (2)$$

$$f(x) = \sum_{w=1}^{52} \left[(q_w h_w \eta_w \alpha_w + \gamma_w) - \sum_{t=1}^T d_t x_{tw} \right] \quad (3)$$

sujeito a:

$$\sum_{t=1}^T d_t x_{tw} \leq C_w \quad (4)$$

$$\sum_{w=1}^p x_{tw} = 1, \quad \forall p = \{2, 3, \dots, 52\} \quad (5)$$

$$x_{tw} - x_{t(w+np)} = 0, \quad \forall n \in N^* \quad (6)$$

$$x_{tw} - x_{(t+n)w} = 0, \quad \forall n \in N^* \quad (7)$$

$$x_{tw} \in \{0, 1\}, \forall t = \{1, 2, \dots, T\}, w = \{1, 2, \dots, 52\} \quad (8)$$

d	p	w _s	w ₁	w ₂	w ₃	w ₄	...	w ₅₂
d ₁	1	w _{s1}	x _{1,1}	x _{1,2}	x _{1,3}	x _{1,4}	...	x _{1,52}
...	1
d _a	1	w _{sa}	x _{a,1}	x _{a,2}	x _{a,3}	x _{a,4}	...	x _{a,52}
d _{a+1}	2	w _{s(a+1)}	x _{a+1,1}	x _{a+1,2}	x _{a+1,3}	x _{a+1,4}	...	x _{a+1,52}
...	2
d _b	2	w _{sb}	x _{b,1}	x _{b,2}	x _{b,3}	x _{b,4}	...	x _{b,52}
d _{b+1}	3	w _{s(b+1)}	x _{b+1,1}	x _{b+1,2}	x _{b+1,3}	x _{b+1,4}	...	x _{b+1,52}
...	3
d _c	3	w _{sc}	x _{c,1}	x _{c,2}	x _{c,3}	x _{c,4}	...	x _{c,52}
...
d _T	p _T	w _{sT}	x _{T,1}	x _{T,2}	x _{T,3}	x _{T,4}	...	x _{T,52}
C _{1..52}			C ₁	C ₂	C ₃	C ₄	...	C ₅₂
D _{1..52}			D ₁	D ₂	D ₃	D ₄	...	D ₅₂
R _{1..52}			R ₁	R ₂	R ₃	R ₄	...	R ₅₂

Fig. 3 – Generalização para o modelo de programação de tarefas.

A função objetivo, dado por (2) e (3), representa a minimização do *ripple* de programação, i.e., capacidade de execução das tarefas menos a demanda programada. Esta minimização é importante para que as equipes de execução possam atender às demandas existentes sem violar diversos argumentos de custos, leis trabalhistas etc.

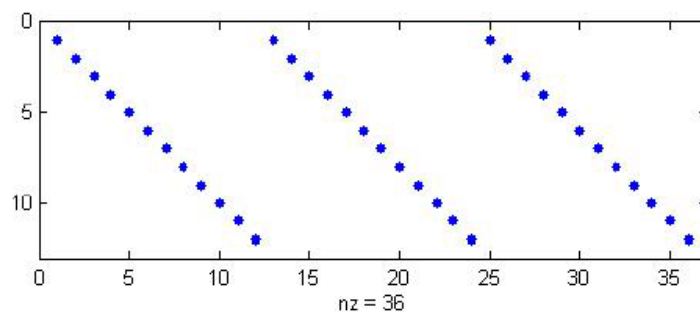


Fig. 4 – Matriz esparsa para a primeira função de restrição do exemplo.

Para exemplificar as funções de restrição do modelo, um exemplo constituído de três tarefas, sendo duas destas com períodos iguais a quinze dias e uma com vinte e um dias, programadas para um período total de doze semanas foi elaborado. A figura 4 representa a matriz esparsa para a primeira função de restrição do modelo matemático, i.e., restrição de atendimento semanal das demandas das tarefas programadas (4). Tais demandas semanais não devem superar o valor da carga líquida C_w semanal.

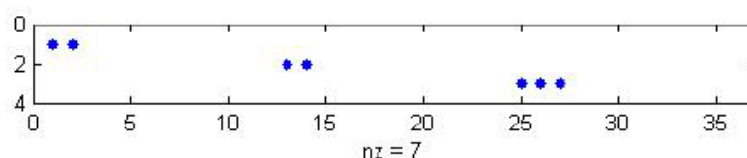


Fig. 5 – Matriz esparsa para a segunda função de restrição do exemplo.

A segunda função de restrição do modelo matemático (5), refere-se ao cumprimento de uma determinada tarefa dentro do intervalo do seu período de programação. A figura 5 ilustra o modelamento desta restrição. Nesta figura, somente uma tarefa deverá ser programada dentro do intervalo do período relativo à tarefa em questão.

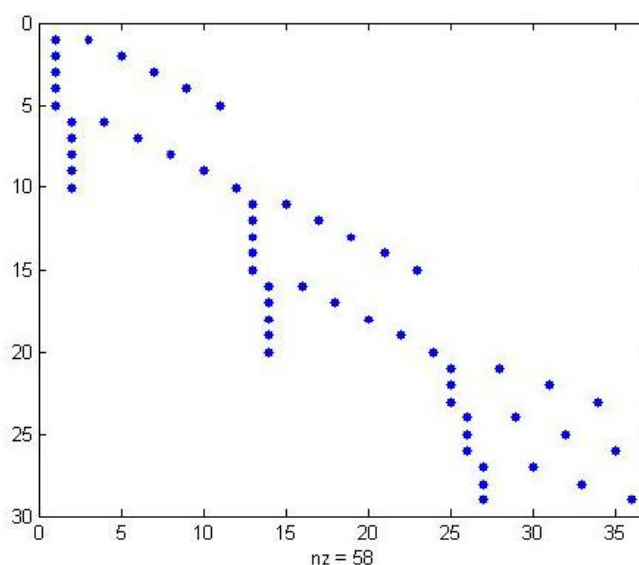


Fig. 6 – Matriz esparsa para a terceira função de restrição do exemplo.

Para a terceira função de restrição do modelo matemático (6), esta refere-se à condição de repetibilidade da função de restrição (5) em todo o período de programação das tarefas, que no caso do exemplo em questão é igual a doze semanas. A figura 6 ilustra o modelamento para a terceira restrição do modelo matemático para as três tarefas do exemplo.

A quarta e última restrição do modelo matemático (7), refere-se a interdependência entre tarefas, i.e., a necessidade que tarefas distintas e de mesma periodicidade tenham que ser programadas juntas e executadas na mesma semana. A figura 7 ilustra esta restrição para as duas tarefas de periodicidade igual a quinze dias.

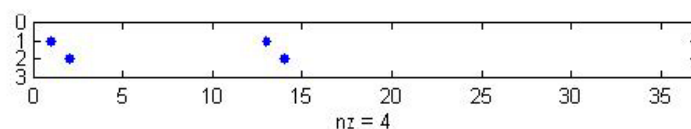


Fig. 7 – Matriz esparsa para a quarta função de restrição do exemplo.

A figura 8 representa a matriz esparsa para todas as funções de restrição do modelo matemático. Esta matriz é utilizada para a solução do modelo através do software Gurobi.

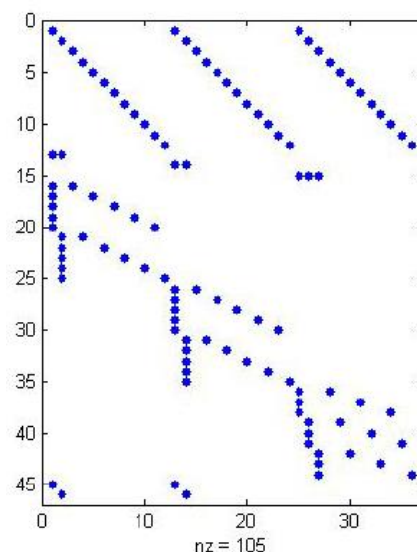


Fig. 8 – Matriz esparsa para todas as função de restrição do exemplo.

4. RESULTADOS

Os resultados obtidos através dos modelos matemáticos de otimização combinatória aplicados ao problema de programação de tarefas de manutenção são apresentados pelas seções A, B, C, D, E e F. O algoritmo desenvolvido para solucionar o PLIB foi executado pelo MatLab juntamente com o Gurobi. O banco de dados com as informações de demandas (Hh) e periodicidades das tarefas de manutenção foi utilizado em conjunto com o algoritmo PLIB para solucionar o problema proposto.

A. Equipe Executante

A figura 9 apresenta o perfil de variação da quantidade da equipe de execução das tarefas durante o período de 52 semanas. Esta variação da quantidade é devida às férias, afastamentos médicos, folgas e deslocamentos locais entre equipes.

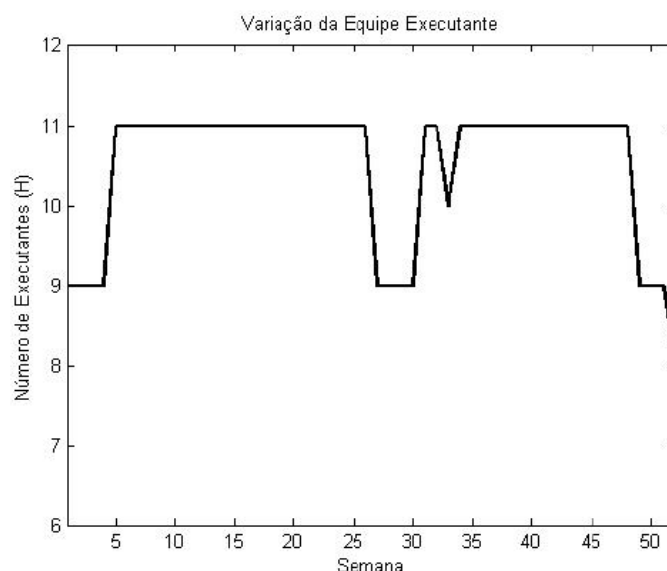


Fig. 9 – Variação da quantidade da equipe de execução das tarefas.

B. Horas Semanais

As horas semanais disponíveis para que as tarefas sejam programadas estão representadas pela figura 10. A variação de horas é devido basicamente aos feriados e compensações destes presentes no ano. A descrição dos feriados e compensações estão conforme a seguir:

- feriado 1 janeiro: ano novo;
- feriado 5 março: carnaval;
- feriado 19 abril: paixão de Cristo;
- feriado 1 maio: dia do trabalho;
- feriado 20 junho: corpus Christi;
- compensação do feriado 7 setembro: independência do Brasil;
- compensação do feriado 12 outubro: nossa senhora aparecida;
- compensação do feriado 2 novembro: finados;
- feriado 15 novembro: proclamação da república;
- feriado 25 dezembro: natal.

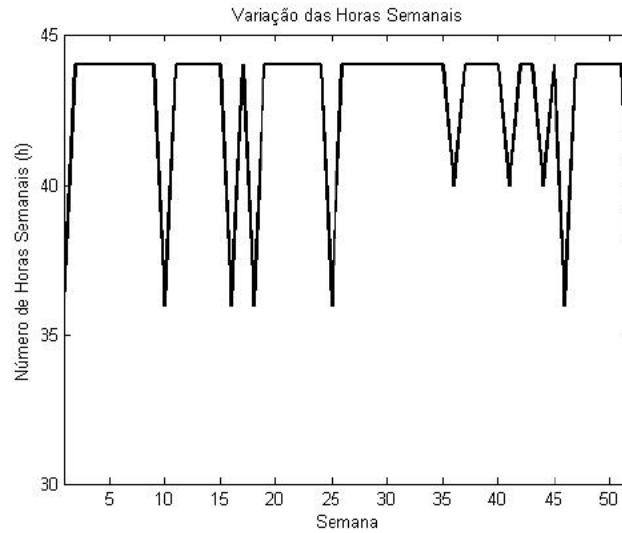


Fig. 10 – Variação da quantidade de horas semanais disponíveis.

C. Carga Líquida

A carga líquida C_w ,

$$C_w = q_w h_w \eta_w \alpha_w + \gamma_w \quad (9)$$

representa a demanda Hh disponível para a tarefa de programação. Contratações de equipes terceiras, percentuais de horas extras e fator de eficiência de trabalho da equipe são alguns dos fatores que influenciam diretamente no valor da carga líquida disponível para a programação das tarefas. Os parâmetros e alguns valores utilizados para cálculo de C_w são apresentados a seguir:

- q_w : quantidade nominal de executantes na semana;
- h_w : horas médias semanais;
- η_w : eficiência de trabalho das equipes executantes. Valor fixo igual a 0.7;
- α_w : fator de horas extras adicionais semanais. Valor fixo igual a 1.10;
- γ_w : homem-hora semanal adicionais contratados de terceiros. Valor fixo igual a 8 Hh.

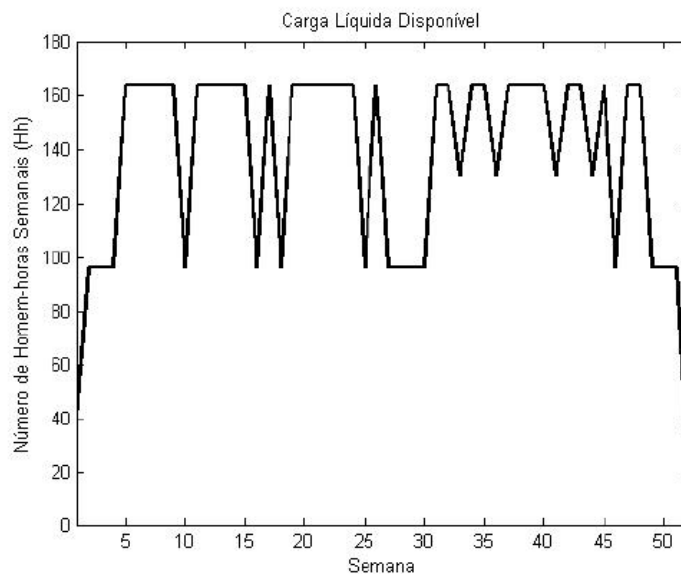


Fig. 11 – Variação da carga líquida semanal.

D. Programação Não Otimizada

A figura 12 apresenta a demanda de Hh para a programação não otimizada das tarefas de manutenção. Esta demanda é resultante caso todas as tarefas fossem inicializadas na primeira semana. O total de recurso da equipe de execução para atender esta demanda não otimizada seria em torno de 754 Hh. Na figura, este valor é igual a 537 Hh, pois as demandas semanais que totalizam 217 Hh foram excluídas da carga líquida pelo uso das simplificações adotadas.

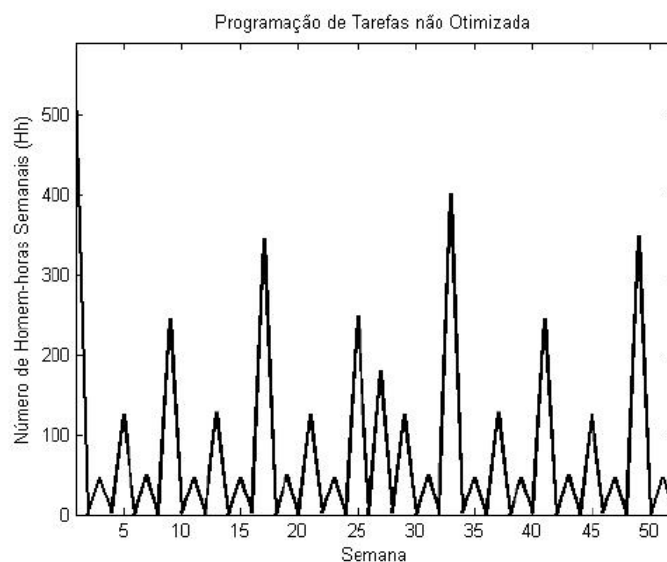


Fig. 12 – Demanda para a programação de tarefas não otimizada.

E. Programação Otimizada

A demanda para a programação otimizada das tarefas de manutenção é apresentada pela figura 13. Tal demanda é resultante da programação otimizada das tarefas visualizadas pela matriz

esparsa dada pela figura 14. Pode ser observado que o pico de demanda para a programação otimizada é inferior a 120 Hh.

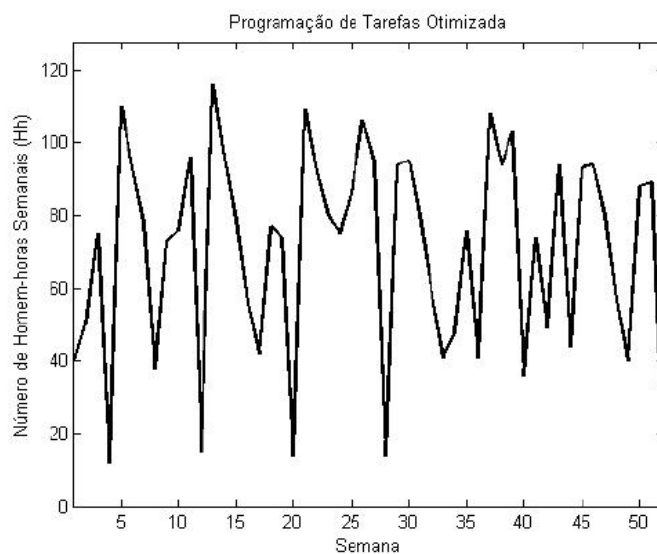


Fig. 13 – Demanda para a programação de tarefas otimizada.

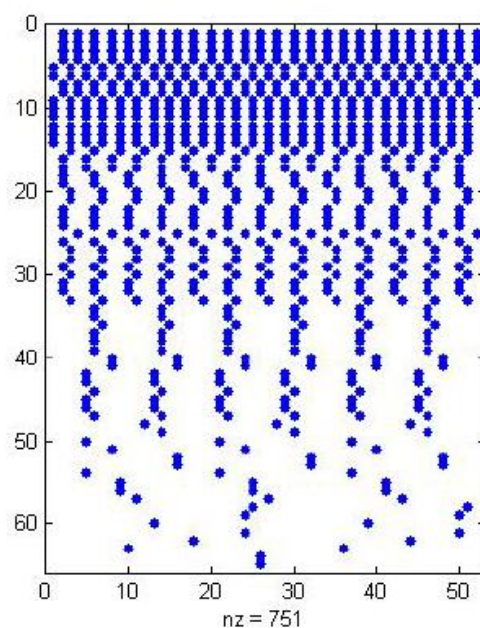


Fig. 14 – Variação da quantidade de horas semanais disponíveis.

F. Resultados Globais

A figura 15 apresenta os resultados para a carga líquida e demandas para as programações otimizadas e não otimizadas semanais.

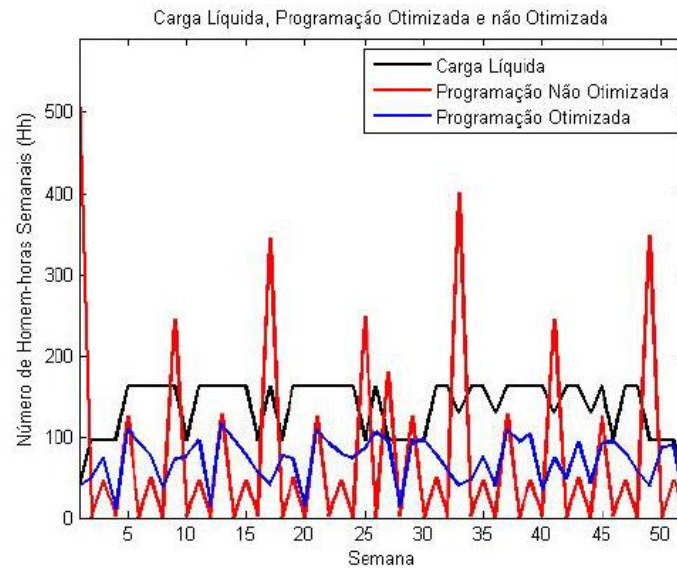


Fig. 15 – Carga líquida, demandas otimizadas e não otimizadas semanais.

5. CONCLUSÃO

Os resultados obtidos através da utilização dos modelos de otimização combinatória operacionalizados pelas ferramentas MatLab em conjunto com Gurobi, possibilitaram a determinação de uma solução exata para o problema de programação de tarefas de manutenção. A redução do *ripple* de programação possibilitou um decréscimo do valor do desvio padrão da demanda não otimizada de $\delta_0 = 120.07$ para um valor de $\delta_1 = 28.72$ referente ao desvio padrão da demanda otimizada.

As simplificações do modelo matemático, através da exclusão das periodicidades semanais, possibilitou maior velocidade para construção das matrizes de restrições, visto que aproximadamente 217 Hh são relativos às demandas de tarefas semanais.

As variações nos parâmetros da carga líquida para análise da factibilidade da solução, especificamente na variação da quantidade da equipe executante q_w , possibilitou um estudo de dimensionamento ótimo da equipe onde o ganho foi em torno de 6 executantes. A equipe inicial era dimensionada para $q_0 = 18$ executantes e a solução do modelo convergiu para uma equipe limite de $q_1 = 12$ executantes.

O esforço computacional também foi muito baixo uma vez que o tempo total para encontrar a solução ótima através do PLIB (MatLab + Gurobi) foi igual a $t = 0.031s$, com um número de iterações igual a $\#it = 71$. O tempo de processamento em conjunto com a obtenção da solução exata para o problema de otimização de programação de tarefas de manutenção faz desta abordagem uma solução superior comparada com típicas soluções aproximadas providas em diversas ferramentas comerciais atuais.

Como melhorias ao modelo PLIB e ao trabalho, pode-se citar os seguintes:

- Modificar o modelo para análise de múltiplas equipes de execução;
- Análise da programação das tarefas dentro da semana;
- Análise dinâmica do modelo prevendo novas tarefas.

6. REFERÊNCIAS:

- [1] A. Cavallo, R. Setola, F. Vasca. Using MATLAB, SIMULINK and Control System Toolbox - A Practical Approach. Prentice Hall Europe, 1996.
- [2] A. G. Cunha, R. Takahashi, C. H. Antunes. Manual de Computação Evolucionária e Metaheurística. 1a ed, Editora UFMG, 2013.
- [3] A. K. S. Jardine. Maintenance, Replacement and Reliability. 1. ed. Ontario, Canada: Pitman Publishing, 1973. 199 p.
- [4] A. K. S. Jardine, A. H. C. Tsang. Maintenance, Replacement and Reliability - Theory and Applications. 2.ed. Boca Raton, FL, USA: CRC Press, 2013. 330 p.
- [5] B. S. Dhillon. Maintainability, Maintenance and Reliability for Engineers. 1. ed. Florida, USA: Taylor & Francis Group, 2006. 214 p.
- [6] C. E. Ebeling. An Introduction to Reliability and Maintainability Engineering. 1. ed. New York, USA: The McGraw-Hill Companies, Inc., 1997. 486 p.
- [7] E. C. Fitch. Proactive Maintenance for Mechanical Systems. 1. ed. Oklahoma, USA: Elsevier, 1992. 337 p.
- [8] E. G. Carrano, L. S. Batista. Otimização em Redes. Notas de Aula, PPGEE/UFMG, 2019.
- [9] E. Zio. An Introduction to the Basics of Reliability and Risk Analysis. 1. ed. 5 Toh Tuck Link, Singapore: World Scientific Publishing Co. Re. Ltd., 2007. 222 p.
- [10] F. G. Guimarães. Multiobjective Optimization Class Notes. Universidade Federal de Minas Gerais. Course of Multiobjective Optimization - EEE910.
- [11] F. G. Guimarães, L. S. Batista, E. G. Carrano. Introduction to Multiobjective Optimization - Class Notes. Universidade Federal de Minas Gerais.
- [12] H. R. G. Viana. Planejamento e Controle da Manutenção. 1. ed. Rio de Janeiro, Brasil: Qualitymark Editora Ltda., 2006. 192 p.
- [13] H. Wang, H. Pham. Reliability and Optimal Maintenance. 1. ed. London, UK: Springer-Verlag, 2006. 345 p.
- [14] J. A. Ramírez, F. Campelo, F. G. Guimarães, L. S. Batista, R. H. C. Takahashi. Class Notes in Optimization. Universidade Federal de Minas Gerais.
- [15] J. D. Patton. Maintainability and Maintenance Management. 1. ed. North Carolina, USA: Instrument Society of America, 1980. 441 p.
- [16] J. S. Arora. Introduction to Optimum Design. 3a ed, Waltham, MA, USA: Elsevier, 2012. 880 p.
- [17] Mathworks. Disponível em: <<http://www.mathworks.com/help/matlab/ref/>>. Último acesso em 05 de julho de 2019.
- [18] Mathworks. MatLab: Optimization toolbox user's guide - version 2. Natick, MA, USA, 2003. 352 p.

- [19] M. Arenales, V. Armentano, R. Morabito, H. Yanasse. Pesquisa Operacional para Cursos de Engenharia. Editora Campus/Elsevier, 2007.
- [20] M. C. Goldbarg, H. P. Luna. Otimização Combinatória e Programação Linear - Modelos e Algoritmos. 2a ed, Editora Campus/Elsevier, 2005.
- [21] M. S. Bazarra, H. D. Sherali, C. M. Shetty. Nonlinear Programming - Theory and Algorithms. 3rd ed. John Wiley & Sons Inc, 2006.
- [22] N. B. Bloom. Reliability Centered Maintenance: Implementation made simple. 1. ed. New York, USA: McGraw-Hill Books, 2005. 291 p.
- [23] S. Boyd, L. Vandenberghe. Convex Optimization. 7. ed. The Edinburgh Building, Cambridge, CB2 8RU, UK: Cambridge University Press, 2009. 716 p.
- [24] R. E. Barlow, L. Hunter. Optimum preventive maintenance policies. Operations Research, v. 8, n. NE, p.90-100, 1960.
- [25] R. E. Barlow. Engineering Reliability. 1. ed. Philadelphia, USA: ASASIAM, 1998. 199 p.
- [26] R. H. C. Takahashi. Scalar and Vectorial Optimization - Class Notes. Universidade Federal de Minas Gerais. Departamento de Matemática, 2007.
- [27] R. K. Mobley. An Introduction to Predictive Maintenance. 2. ed. Woburn, MA, USA: Butterworth-Heinemann Ltd, 2002. 438 p.
- [28] R. R. Saldanha. Optimization in Electric Engineering - Class Notes. Universidade Federal de Minas Gerais. Course of Optimization in Electric Engineering - EEE948, 2014.
- [29] S. M. Sinha. Mathematical Programming - Theory and Methods. Elsevier Science & Technology Books, 2006.
- [30] S. S. Rao. Engineering Optimization - Theory and Practice. 4th ed. John Wiley & Sons Inc, 2009.
- [31] Wikipedia: Set Mathematics. Disponível em: <[https://en.wikipedia.org/wiki/Set mathematics](https://en.wikipedia.org/wiki/Set_mathematics)>. Último acesso em 04 de julho de 2019.