

JOSÉ ANTÓNIO PEREIRA GONÇALVES

Mestrado/Licenciatura em Nome do Grau Anterior

APLICAÇÃO DE AGENDAMENTO SOBRE EXAMES PET/CT

ALGUMAS CONSIDERAÇÕES SOBRE A VIDA,
O UNIVERSO E TUDO O MAIS

MESTRADO EM ENGENHARIA E GESTÃO INDUSTRIAL

Universidade NOVA de Lisboa

Draft: 4 de setembro de 2025

APLICAÇÃO DE AGENDAMENTO SOBRE EXAMES PET/CT

ALGUMAS CONSIDERAÇÕES SOBRE A VIDA,
O UNIVERSO E TUDO O MAIS

JOSÉ ANTÓNIO PEREIRA GONÇALVES

Mestrado/Licenciatura em Nome do Grau Anterior

Orientadora: Mary Doe Adviser Name
Full Professor, NOVA University Lisbon

Coorientadores: John Doe Co-Adviser Name
Associate Professor, NOVA University Lisbon
John Doe other Co-Adviser Name
Full Professor, NOVA University Lisbon

MESTRADO EM ENGENHARIA E GESTÃO INDUSTRIAL

Universidade NOVA de Lisboa

Draft: 4 de setembro de 2025

RESUMO

Independentemente da língua em que a dissertação está escrita, geralmente esta contém pelo menos dois resumos: um resumo na mesma língua do texto principal e outro resumo numa outra língua.

A ordem dos resumos varia de acordo com a escola. Se a sua escola tiver regulamentos específicos sobre a ordem dos resumos, o template (L^AT_EX) NOVAthesis L^AT_EX (*novathesis*) irá respeitá-los. Caso contrário, a regra padrão no template *novathesis* é ter em primeiro lugar o resumo *no mesmo idioma do texto principal* e depois o resumo *no outro idioma*. Por exemplo, se a dissertação for escrita em português, a ordem dos resumos será primeiro o português e depois o inglês, seguido do texto principal em português. Se a dissertação for escrita em inglês, a ordem dos resumos será primeiro em inglês e depois em português, seguida do texto principal em inglês. No entanto, esse pedido pode ser personalizado adicionando um dos seguintes ao arquivo `5_packages.tex`.

```
\abstractorder(<MAIN_LANG>):={<LANG_1>,\dots,<LANG_N>}
```

Por exemplo, para um documento escrito em Alemão com resumos em Alemão, Inglês e Italiano (por esta ordem), pode usar-se:

```
\ntsetup{abstractorder={de={de,en,it}}}
```

Relativamente ao seu conteúdo, os resumos não devem ultrapassar uma página e frequentemente tentam responder às seguintes questões (é imprescindível a adaptação às práticas habituais da sua área científica):

1. Qual é o problema?
2. Porque é que é um problema interessante/desafiante?
3. Qual é a proposta de abordagem/solução?
4. Quais são as consequências/resultados da solução proposta?

Palavras-chave: Primeira palavra-chave, Outra palavra-chave, Mais uma palavra-chave, A última palavra-chave

ABSTRACT

Regardless of the language in which the dissertation is written, usually there are at least two abstracts: one abstract in the same language as the main text, and another abstract in some other language.

The abstracts' order varies with the school. If your school has specific regulations concerning the abstracts' order, the `novathesis` (L^AT_EX) template will respect them. Otherwise, the default rule in the `novathesis` template is to have in first place the abstract in *the same language as main text*, and then the abstract in *the other language*. For example, if the dissertation is written in Portuguese, the abstracts' order will be first Portuguese and then English, followed by the main text in Portuguese. If the dissertation is written in English, the abstracts' order will be first English and then Portuguese, followed by the main text in English. However, this order can be customized by adding one of the following to the file `5_packages.tex`.

```
\ntsetup{abstractorder={<LANG_1>,...,<LANG_N>}}  
\ntsetup{abstractorder={<MAIN_LANG>={<LANG_1>,...,<LANG_N>}}}
```

For example, for a main document written in German with abstracts written in German, English and Italian (by this order) use:

```
\ntsetup{abstractorder={de={de,en,it}}}
```

Concerning its contents, the abstracts should not exceed one page and may answer the following questions (it is essential to adapt to the usual practices of your scientific area):

1. What is the problem?
2. Why is this problem interesting/challenging?
3. What is the proposed approach/solution/contribution?
4. What results (implications/consequences) from the solution?

Keywords: One keyword, Another keyword, Yet another keyword, One keyword more, The last keyword

ÍNDICE

Índice de Figuras	v
List of Algorithms	vii
Siglas	viii
1 Introdução	1
1.1 Motivação e Contexto	1
1.2 Apresentação do Problema e Objetivo	1
1.3 Abordagem Metodológica	2
2 Base Teórica	3
2.1 Extensões, Restrições, e Critérios	3
2.2 Abordagens para a Resolução	4
2.2.1 Algoritmos Exatos	4
2.2.2 Heurísticas	5
2.2.3 Meta-Heurísticas	5
3 Revisão de Literatura	7
3.1 Abordagens para a Resolução	7
3.1.1 Algoritmos Exatos	7
3.1.2 Heurísticas	8
3.1.3 Meta-Heurísticas	8
4 Mapeamento dos Exames	10
5 Desenvolvimento dos Modelos	12
5.1 Problema de <i>Makespan</i>	12
5.1.1 Modelo 1	15
5.1.2 Modelo 2	21
5.1.3 Modelo 3	25

6	Resultados	31
6.1	Problema de <i>Makespan</i>	31
6.1.1	Modelo 1	32
6.1.2	Modelo 2	34
6.1.3	Modelo 3	36
6.1.4	Comparação entre modelos	37
6.1.5	Comparação com instâncias de <i>benchmark</i>	39
	Bibliografia	41
	Apêndices	
.1	Appendix:Tabelas	45
.2	Appendix:Imagens	56

ÍNDICE DE FIGURAS

5.1	Impacto das diferentes formas de calcular C_{\max}	16
5.2	Impacto das duas forma de gerar t sobre a procura da vizinhança.	16
5.3	Processo de transição da solução s para a solução s' para o modelo 1.	17
5.4	Evolução do valor da função objetivo ao longo da otimização do problema <i>makespan</i> com o modelo 1, para a obtenção destas figuras, as variáveis utilizadas foram: $L_k = 5000$, $CP = 100$, $\alpha = 0.8$, $P = 1000$, $p_0 = 0.9$	20
5.5	Impacto do recalculo de ini e fim sobre os instantes considerados na próxima iteração.	21
5.6	Processo de transição da solução s para a solução s' para o modelo 2.	22
5.7	Evolução do valor da função objetivo ao longo da otimização do problema <i>makespan</i> com o modelo 2, para a obtenção destas figuras, as variáveis utilizadas foram: $L_k = 500$, $CP = 100$, $\alpha = 0.9$, $P = 0$, $p_0 = 0.9$	24
5.8	Impacto do recalculo de ini e fim sobre a lista de candidatos da próxima iteração.	25
5.9	Exemplo da solução 3-4-1-2	28
5.10	Exemplo da solução 2-3-1-4	28
5.11	Processo de transição da solução s para a solução s' para o modelo 3.	29
5.12	Evolução do valor da função objetivo ao longo da otimização do problema <i>makespan</i> com o modelo 3, para a obtenção destas figuras, as variáveis utilizadas foram: $L_k = 500$, $CP = 100$, $\alpha = 0.9$, $P = 0$, $p_0 = 0.9$, <i>left-shift timetabling</i> , geração de vizinhos através de troca.	29
6.1	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 1.	33
6.2	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 2.	35
6.3	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 3 com <i>left shifting</i>	36

.4	Gráfico de contorno das cinco variáveis, relativamente ao <i>makespan</i> , do modelo 1 do problema 1	46
.5	Gráfico de contorno das cinco variáveis, relativamente ao tempo, do modelo 1 do problema 1	48
.6	Gráfico de contorno das quatro variáveis relevantes, relativamente ao <i>makespan</i> , do modelo 2 do problema 1	50
.7	Gráfico de contorno das quatro variáveis relevantes, relativamente ao tempo, do modelo 2 do problema 1	52
.8	Gráfico de contorno das quatro variáveis relevantes, relativamente ao <i>makespan</i> , do modelo 3 do problema 1	54
.9	Gráfico de contorno das quatro variáveis relevantes, relativamente ao tempo, do modelo 3 do problema 1	56
.10	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 1.	57
.11	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 2.	58
.12	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 3 com <i>left shifting</i>	58
.13	Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 3 com <i>enhanced left shifting</i>	59

LIST OF ALGORITHMS

1	Pseudo-código de gerção de novos vizinhos, a sua avaliação, aceitação ou rejeição. Modelo 1.	18
2	Pseudo-código que gera os candidatos durante a criação da vizinhança. .	22
3	Pseudo-código de geração de novos vizinhos, a sua avaliação, aceitação ou rejeição. Modelo 2.	23
4	Pseudo-código do algoritmo para encontrar o instante mais cedo possível.	26
5	Pseudo-código de <i>left-shift timetabling</i>	26
6	Pseudo-código de <i>enhanced left-shift timetabling</i>	26

SIGLAS

novathesis NOVAthesis L^AT_EX (*pp. [i](#), [ii](#)*)

INTRODUÇÃO

Este capítulo irá introduzir o tema desta dissertação, a sua estrutura, a motivação e contexto, os objetivos a atingir e a abordagem tomada.

1.1 Motivação e Contexto

O custo associado ao setor de saúde tem vindo a aumentar, com os países do G7 a pagar mais de 10% do seu GDP para este fim. O elevado custo deve-se a vários fatores, ao envelhecimento da população, à degradação do estilo de vida, ao aumento da estadia em hospital, entre outros [1].

A realização de exames é um das principais atividades de um hospital, como forma de apoio ao diagnóstico e à cirurgia. Em Portugal existe por isso tempos máximos de resposta garantidos [2], medicina nuclear (MN) insere-se no meio complementar de diagnóstico e terapêutica, devendo ocorrer um exame desta natureza em menos de 30 dias a partir da indicação clínica. Contudo, frequentemente estes limites não são cumpridos, devido à maior procura destes meios [3] e a ineficiências na realização desta atividade.

Lean Healthcare apresenta-se como uma das soluções para combater as ineficiências presentes no setor de saúde, para tal existe uma grande variedade de abordagens mas também muitos desafios na implementação de soluções. Não sei se deva referir *Lean Healthcare*.

1.2 Apresentação do Problema e Objetivo

Esta dissertação teve como *pich* a melhoria da eficiência do departamento de MN do Hospital Garcia de Orta que, a priori, seria através da melhoria do agendamento de pacientes ao longo do dia de trabalho de forma a realizar um número maior de exames com os recursos humanos e físicos já existentes.

Cada exame é composto por várias tarefas, na qual podem ser necessários vários recursos

diferentes, não ocorrendo tempo de espera entre tarefas adjacentes de um mesmo exame. Esta dissertação pretende alcançar três objetivos:

1. O mapeamento dos exames mais frequentemente realizados, com possibilidade de mapear os exames que ocorrem com menor frequência;
2. A criação de ferramentas de apoio à decisão, duas em contexto operacional e outra para contexto tática;
3. A validação de resultados obtidos pelas ferramentas em termos de viabilidade e robustez.

1.3 Abordagem Metodológica

De forma a atingir os objetivos previamente referidos é necessário estruturar a abordagem metodológica. Será realizada uma revisão de literatura que abordará os tipos de problemas de agendamento e os respetivos métodos de resolução. De seguida serão mapeados e validados os exames, e posteriormente recolhidos dados relativos à duração de cada atividade. Seguidamente serão desenvolvidos os modelos de apoio à decisão já mencionados, com aplicação destes com os dados do sistema real com apresentação dos resultados obtidos, mas também a comparação de instâncias *benchmark* existentes na literatura. Finalmente serão propostos trabalhos futuros, e serão retiradas conclusões.

BASE TEÓRICA

Este capítulo irá introduzir alguns conhecimentos necessários para comunicar eficientemente a revisão da literatura e o trabalho realizado de forma geral.

2.1 Extensões, Restrições, e Critérios

O problema aqui apresentado poderá ser englobado na família de problemas *Job-Shop*, um dos problemas mais estudados na área de investigação operacional.

Este problema é definido por um conjunto finito J de trabalhos, um conjunto finito R de recursos, e um conjunto finito O de operações. Cada trabalho j é uma sequência de n_j operações consecutivas em O , em que cada operação $i \in O$ será processada no recurso R_i com duração contínua de τ_i e sem antecipação de operações [4].

Contudo, o problema em estudo não será apenas descrito pela definição clássica de *Job-Shop*, existem então uma grande variedade de extensões, restrições, e critérios que permitem modificar o problema base de forma a ser mais fidedigno o sistema em estudo. Para tal, o problema em estudo será descrito como *Flexible Multi-Resource Job Shop with No-Wait*.

A extensão *Flexible* foi descrito pela primeira vez por Brucker & Schlie [5] e dado o nome de *Flexible* por Brandimarte [6], esta adição é uma das mais importantes na generalização do problema *Job-Shop* para o contexto real, existe no entanto o aumento da complexidade e a impossibilidade de utilizar o método tradicional na sua resolução [4]. Existe, para cada operação $i \in O$ um conjunto de recursos $R_i \subseteq R$ que podem ser utilizados, ou seja, existe um conjunto de recursos-tipo que exercem a mesma função, com duração de τ_i^k para o recurso $k \in R_i$.

A extensão *Multi-Resource* foi descrito pela primeira vez por Brucker & Neyer [7] e por Dauzère-Pérès et al. [8] na qual existe, para cada operação $i \in O$, $m(i)$ recursos diferentes, tal que o conjunto R_i^k contem o k recurso-tipo da operação O_i , ou seja, cada operação terá

de ser processada por $m(i)$ recursos-tipo.

A restrição *Minimum/Maximum Time Lag* limita o tempo decorrido entre operações sequenciais de um trabalho, sendo *No-Wait* o caso extremo desta restrição, garantindo que não existe tempo de espera entre operações sequenciais, este conceito foi proposto por Callahan [9], tal que, o começo da operação O_{i+1} ocorra no mesmo momento que a operação O_i termina, caso estas façam parte do mesmo trabalho j .

Ao mesmo tempo também é possível resolver o problema apresentado de forma a maximizar ou minimizar diferentes critérios. Estes podem ser baseado em tempo, número de trabalhos, custo, receita, ou impacto no meio ambiente.

2.2 Abordagens para a Resolução

Existem três categorias gerais para a resolução de problemas *Job-Shop*, algoritmos exatos, heurísticas, e meta-heurísticas [10]. De seguida serão apresentados alguns métodos de resolução do problema *Job-shop* e uma breve descrição de como funcionam.

2.2.1 Algoritmos Exatos

Historicamente, a utilização de métodos exatos foi a principal abordagem utilizada, com especial ênfase nos artigos iniciais sobre o problema clássico de *Job-Shop*. A utilização de *Integer Linear Programming* (ILP), *Mixed Integer Linear Programming* (MILP) são as principais abordagens utilizadas.

A forma canónica de *Linear Programming* pode ser expressa por:

$$\text{Maximizar } z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeito a:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$x_j \geq 0, j = 1, \dots, n$$

Existem vários modelos utilizados para a representar o sistema em estudo, o modelo disjuntivo, o modelo indexado no tempo, e o modelo baseado em ordenação.

O modelo disjuntivo utiliza um grafo denominado $G(V, C \cup D)$, em que V é o conjunto de vértices e que representam as operações dos vários trabalhos, C é o conjunto de arcos

conjuntivos que ligam as operações i e $i + 1$ de um trabalho, D é o conjunto de arcos disjuntivos que ligam as operações que requerem o mesmo recurso. Desta forma é escolhida a ordem de processamento em cada recurso de forma a cumprir com as restrições e minimizando/maximizando o critério em estudo.

O modelo indexado no tempo, por sua vez, procura definir o instante de início de cada operação, sendo esta a variável de decisão que o modelo utilizará. O modelo baseado em ordenação define explicitamente a sequência de operações que deverão ocorrer em cada recurso.

Além do modelo, também será necessário um programa capaz de o resolver, como o Gurobi e o CPLEX, que utilizam algoritmos para encontrar a solução ótima. Em contra mão, é possível desenvolver algoritmos específicos a cada problema. Existem vários algoritmos utilizados para este fim, como *Branch and Bound* ou *Branch and Cut*. Estes algoritmos são utilizados para reduzir o número de soluções a explorar ao não considerar soluções que certamente serão piores.

2.2.2 Heurísticas

Heurísticas são algoritmos de rápida resolução que devem permitir a obtenção de uma solução viável de forma rápida, por isso será de esperar que esta abordagem não resulte na obtenção da solução ótima. Frequentemente a formulação de uma heurística apenas funcionará para o problema para qual foi desenhada, devido à intuição apresentada na sua construção [11].

2.2.3 Meta-Heurísticas

Meta-heurísticas são o meio termo entre a qualidade das soluções apresentadas por algoritmos exatos e a rapidez das heurísticas. Existe uma grande quantidade de abordagens existentes, podem ser baseadas em população, em trajetórias, com memória ou sem, ou baseadas em processos naturais.

Algumas das meta-heurísticas com mais impacto são: *Simulated Annealing*, *Differential Evolution*, *Particle Swarm Optimization*, *Genetic Algorithm*, *Ant Colony Optimization*, e *Tabu Search Algorithm* [11]. Todas estas meta-heurísticas poderão ser utilizadas para resolver o problemas apresentado e serão discutidas na próxima secção.

Simulated Annealing é uma meta-heurística evolucionária baseada no processo de arrefecimento de metais, em que se verifica uma menor energia do sistema quando este arrefecimento acontece lentamente. Desta forma, é gerada uma solução i com uma vizinhança S_i de onde é gerada uma nova solução j , que será avaliada consoante a função

objetivo $f(j)$, a solução j poderá ser aceite mesmo não apresentando melhor valor, , sendo esta a principal diferenciação de outras meta-heurísticas, de acordo com o algoritmo Metropolis [12]:

$$Pr\{\text{aceitar } j\} = \begin{cases} 1, & \text{se } f(j) < f(i) \\ e^{-\frac{f(i)-f(j)}{c_k}}, & \text{caso contrário} \end{cases}$$

Onde c_k é a temperatura atual. No início do algoritmo, quando a temperatura é alta, existe probabilidade acrescida de aceitar soluções piores, ao reduzirmos a temperatura com cada iteração, a probabilidade de aceitar soluções piores que a atual diminui. Também será necessário definir qual a temperatura inicial do algoritmo, como se realiza o arrefecimento de c_k , quantas vizinhanças procurar por iteração, e quando se dá por terminado o algoritmo. Cada vizinho visitado pode, ou não, representar uma solução viável, deve-se ter cuidado na escolha da representação da solução e na geração da vizinhança para garantir a viabilidade da solução, ou possivelmente relaxar certas restrições ao incluir a punição sobre a sua violação na função objetivo.

Manter o que está a baixo? Se sim, acho que deveria escrever para todas as meta-heurísticas referidas acima. Mas não sei se deva manter.

Particle Swarm Optimization é uma meta-heurística populacional baseada no comportamento natural de animais de rebanho durante a procura de comida. Para tal são gerados P partículas, tal que a partícula i se encontra na posição $X^i(t)$ na iteração t , bem como uma velocidade $V^i(t)$. Com cada iteração a posição e velocidade de cada partícula é atualizada:

$$\begin{aligned} X^i(t+1) &= X^i(t) + V^i(t+1) \\ V^i(t+1) &= \omega V^i(t) + c_1 r_1 (pbest^i - X^i(t)) + c_2 r_2 (gbest - X^i(t)) \end{aligned}$$

Onde r_1 e r_2 são valores aleatórios entre 0 e 1, ω , c_1 e c_2 são os parâmetros da meta-heurística e chamam-se de inércia, coeficiente cognitivo e coeficiente social, respetivamente. $pbest^i$ é a melhor solução encontrada pela partícula i , $gbest$ é a melhor solução já encontrada.

Tal como SA, não existe a garantia inerente da viabilidade de cada solução, deve-se por isso utilizar um algoritmo que repare cada solução ou relaxar restrições para a função objetivo, com um peso sobre tal. Também será necessário decidir quando terminar a procura de novas soluções, poder-se-à utilizar um conceito semelhante ao descrito anteriormente para SA.

REVISÃO DE LITERATURA

Este capítulo irá explorar a literatura existente em relação às abordagens utilizadas para a resolução do problema *Job-Shop*, com as extensões e restrições discutidas, e outras que além.

3.1 Abordagens para a Resolução

Nesta secção será apresentada uma amostra da literatura existente sobre as abordagens para a resolução dos problemas *Job-Shop* e os seus derivados. A literatura aqui apresentada não será exaustiva, mas demonstrará algumas tendências para problemas próximos ao apresentado nesta dissertação.

O critério mais estudado nesta área de investigação é a minimização do *makespan* [13], mesmo assim existem alguns critérios diferentes que têm aparecido na literatura. Desta forma, pretende-se dar a conhecer alguns destas outras vertentes.

3.1.1 Algoritmos Exatos

Algoritmos exatos foram a principal abordagem utilizada para resolver problemas de *Job-Shop* mas têm vindo a tornar-se menos considerados com o aumento do número de extensões e restrições que os novos problemas apresentam [10].

Como já referido, existem três grandes modelos utilizados neste espaço. O modelo disjuntivo, o modelo indexado no tempo, e o modelo baseado em ordenação. Ku & Beck [14] avaliaram os modelos de acordo com a sua eficiência computacional, onde se observa que o modelo disjuntivo é o mais eficiente a resolver problemas *Job-Shop* de pequenas e grandes dimensões. Será razoável esperar que este facto seja verdade para outras vertentes do problema.

Dauzère-Pérès et al. [4] apresentam uma formulação do modelo disjuntivo utilizando *MILP* para resolver o problema *Flexible Job-Shop* com o objetivo de minimizar o *makespan*, bem como este deve ser modificado de forma a poderem ser adicionadas novas extensões e restrições.

Existem também artigos que geram ou criam problemas que se tornam em *benchmarks*, sendo utilizados em vários artigos de forma a comparar resultados. Um exemplo é o *SFJS* e *MFJS* criado por Fattahi et al. [15], onde também se apresenta uma formulação para resolver o problema *Flexible Job-Shop*. Em Ozbakir et al. [16] é apresentada outra formulação baseada no modelo disjuntivo, ao mesmo tempo é comparada com o modelo de Fattahi et al. [15] em relação à eficiência computacional, verificando que esta formulação é superior na maioria das instâncias. Em Thörnblad et al. [17] é apresentada e comparada uma formulação baseada no modelo indexado no tempo, onde se observa a inferior eficiência deste modelo.

Samarghandi [18] propõe um modelo *MILP* para a resolução do problema *No-wait Job-Shop* com o objetivo de minimizar o *makespan* com a restrição adicional de não ultrapassar a data limite de cada trabalho. Também apresenta dois modelos de *Constraint Programming* que resultam em boas soluções relativamente ao modelo *MILP* em problemas de grande dimensão.

Em Behmanesh et al. [19] é apresentado um modelo *MILP* para resolver o problema *Flexible Multi-Resource Job-Shop*.

3.1.2 Heurísticas

A heurística Nawaz-Enscore-Ham [20] foi desenvolvida para o problema *Flow-Shop*, mas que já foi utilizada em vários artigos, especialmente na criação de soluções iniciais [21], [22] em problemas *Job-Shop*.

Um possível relaxamento do problema em estudo, *Blocking No-Wait and Controlable Times* ou *Blocking No-Wait and Maximum Time Lag*, já foi estudado. Em Aschauer et al. [23] é apresentada uma heurística para o sequenciamento de trabalhos e um algoritmo de *timetabling*, apresentando boas soluções em pouco tempo.

3.1.3 Meta-Heurísticas

Xia & Wu. [24] utilizam um algoritmo híbrido entre *PSO* e *SA* na resolução do problema *Flexible Job-Shop* com tempos de processamento diferentes dentro de cada recurso-tipo. Onde *PSO* é utilizada na atribuição de operações às máquinas, enquanto *SA* é utilizada para escalonar operação a cada recurso.

Sriboonchandr et al. **sriboonchandrImprovedDifferentialEvolution2019** utilizam *Improved Differential Evolution* na resolução do problema *Flexible Job-Shop*, contudo não apresenta a qualidade esperada das soluções.

Jonathan et al. [25] comparam quatro meta-heurísticas para resolver o problema *Flexible Flow-Shop with No-Wait*. A meta-heurística proposta, *Whale Optimization Algorithm* gera solução com qualidade semelhante a *SA*, *Genetic Algorithm* e *Minimum Deviation Algorithm*, contudo esta última demonstra tempo computacional muito menor.

Ying & Lin. [21] comparam quatro variantes de *SA* na resolução do problema *Job-Shop with No-Wait* com minimização do tempo total de conclusão. Demonstram que a utilização adicional do retorno e a existência de várias soluções iniciais permitem a obtenção de melhores soluções no mesmo espaço de tempo.

Perrachon et al. [26] sugerem a utilização de *SA* para a otimização do problema *Flexible Multi-Resource Job-Shop*. Para tal são exploradas oito diferentes estruturas da vizinhança baseadas no modelo disjuntivo. A restrição *Blocking* encontra-se relaxada neste problema, ao ser possível que vários recursos sejam necessários durante fases diferentes de cada operação, podendo ser visto como a decomposição da operação em várias partes com a restrição de *No-Wait* entre elas.

Caumond et al. [27] apresenta uma meta-heurística baseada no modelo disjuntivo, utilizando *Memetic Algorithm* para a resolução do problema *Job-Shop with No-Wait* e *Job-Shop with Maximum Time Lag*.

MAPEAMENTO DOS EXAMES

O processo de mapeamento dos exames começou com a decisão de quais incluir no estudo. Nem todos os exames foram considerados devido à baixa frequência com que estes ocorrem e a consequente falta de dados em relação à duração das suas atividades. Desta forma, apenas foram mapeados aqueles que ocorrem pelos menos mensalmente. De seguida, foi necessário definir os recursos-tipo existentes no sistema, estes podem ser divididos em recursos humanos, técnicos auxiliares de saúde (TAS), enfermeiros, técnicos superiores de diagnóstico e terapêutica de medicina nuclear (TSDT), médicos especialistas (ME), e médicos cardiologistas (MC); e divididos em recursos físicos, gabinetes médicos (GB), sala de espera (SE), sala de espera de crianças (SEC), sala cortinas (SC), sala polivalente (SP), sala enfermagem 1 (SE1), sala enfermagem 2 (SE2), sala enfermagem 3 (SE3), sala câmara gama (SCG), e sala tomografo (ST).

Tabela 4.1: Capacidade dos recursos humanos

Recurso-tipo	TAS	Enfermeiros	TSDT	ME	MC
Capacidade	4	4	4	2	1

Tabela 4.2: Capacidade dos recursos físicos

Recurso-tipo	GM	SE	SEC	SC	SP	SE1	SE2	SE3	SCG	ST
Capacidade	2	4	3	2	1	1	3	3	1	1

Cada recurso-tipo tem uma quantidade de recursos disponíveis, para os recursos humanos corresponde ao número de profissionais em cada grupo existentes no departamento, para os recursos físico corresponde à lotação de pacientes que podem suportar, iremos considerar que existe portanto uma capacidade máxima para cada recurso-tipo, estas capacidades estão apresentadas na Tabela 4.1 e Tabela 4.2, respetivamente. Admite-se que cada recurso atribuído a uma tarefa será utilizado durante a duração total da atividade.

Considerou-se ainda a possibilidade de ter a capacidade dos recursos variável com o tempo, de forma a modelar os diferentes turnos e os períodos onde existe mudança entre

estes. Contudo, com mais discussão decidiu-se que na prática dificilmente seria possível cumprir com uma agenda criada desta forma.

Existem recursos-tipo que podem ser equiparados a outros em certas atividades. A SL e a SE1 podem ser tidas como equivalentes em certas atividades. Por isso foi necessário criar recursos-tipo fictícios. Para o exemplo anterior, foi criado o recurso-tipo SP ou SE1, sempre que uma destas salas é explicitamente necessária, a sua utilização é representada no recurso-tipo correspondente e no recurso-tipo fictício, caso seja indiferente a sala necessária a utilização apenas é representada no recurso-tipo fictício.

A SE2 e a SE3 podem também ser tidas como equiparadas, mas quando a SE2 é explicitamente necessária toda ela é ocupada por uma pessoa, efetivamente diminuindo a sua capacidade de três para uma pessoa. O recurso fictício SE2 ou SE3 continua a ter uma capacidade de seis pessoas. Ambos os casos apresentam-se na Tabela 4.3, onde se observa a capacidade efetiva destes recursos.

Tabela 4.3: Capacidade dos recursos fictícios

Recursos fictícios	SP	SE1	SP ou SE1	SE2	SE2 ou SE3
Capacidade	1	1	2	1	6

Num passo seguinte, cada exame foi decomposto nas suas atividades. Considerou-se que uma nova atividade deve começar quando existe uma mudança dos recursos necessários. Foram então realizadas entrevistas de forma a realizar a decomposição dos exames, no seguimento das entrevistas as atividades foram validadas pelos vários grupos profissionais. Rapidamente identificou-se a necessidade de diferenciar o mapeamento entre pacientes com mobilidade autónoma, e aqueles com mobilidade condicionada, e crianças, existindo diferenças nos recursos necessários e na duração de cada atividade. Seguidamente, as durações de cada atividade foram recolhidas utilizando folhas de registo provenientes do mapeamento realizado. A duração média de cada atividade foi obtida através da hora de início e de fim de cada atividade registadas pelos profissionais, será de salientar que mesmo sendo este um problema de *No-Wait*, esta restrição nem sempre foi verificado nas folhas de registo.

Em certas atividades considera-se que a permanência de um recurso durante a sua totalidade como uma simplificação do problema, podemos utilizar como exemplo a atribuição de um TAS ao tempo de espera decorrido dentro do departamento por cada paciente, contudo não foi possível definir explicitamente quando é que estes recursos seriam necessários, sendo esta uma fonte de variabilidade do sistema, impossibilitando a decomposição da atividade.

Todas as folhas de registo e respetivos dados estarão em anexo com esta dissertação.

DESENVOLVIMENTO DOS MODELOS

Neste capítulo três problemas devem ser resolvidos, para tal foram desenvolvidos vários modelos. No primeiro problema existem recursos e exames fixos, sendo o objetivo minimizar o tempo entre o começo e o fim de todos os exames, chamado de *makespan*. O segundo problema tem recursos e tempo fixos, sendo o objetivo maximizar o número de exames a realizar, utilizando um somatório simples ou ponderado. Finalmente, o terceiro problema é caracterizado por exame e tempos fixos, sendo o objetivo minimizar os recursos necessários para realizar todos os exames no tempo definido.

5.1 Problema de *Makespan*

Este problema irá utilizar o exemplo de uma segunda-feira típica em termos de exames realizados. Ocorrendo 3 cintigrafia tiroideia, 5 cintigrafia pulmonar de ventilação/inalação + perfusão, 10 cintigrafia miocárdica de perfusão em repouso, 1 cintigrafia das glândulas salivares, 10 PET - estudo corpo inteiro com FDG.

A próxima formulação é a que tradicionalmente se encontra quando indexada no tempo, chamá-la-emos de *MILP-trad*:

Conjuntos:

O conjunto de trabalhos $I, i \in I := (1, \dots, n)$

O conjunto de operações $K, k \in K := (1, \dots, K_{\max})$

O conjunto de recursos-tipo $R, p \in R := (1, \dots, R_{\max})$

O conjunto de instantes de tempo $T, t \in T := (1, \dots, T_{\max})$

Parâmetros:

$\rho_{i,k}$ é o tempo de processamento da operação k do trabalho i

$r_{i,p,k}$ é o número de recursos p necessários para realizar a operação k do trabalho i

C_p é a capacidade do recursos do tipo p

Variáveis de Decisão:

$X_{t,i,k}$ é uma variável binária com valor 1 se a operação k do trabalho i ocorrer durante o instante t , caso contrário tem valor 0

$Z_{t,i,k}$ é uma variável binária com valor 1 se a operação k do trabalho i começar no instante t , caso contrário tem valor 0

$S_{i,k}$ é uma variável inteira que representa o valor de começo da operação k do trabalho i

$F_{i,k}$ é uma variável inteira que representa o valor de fim da operação k do trabalho i

C_{\max} é uma variável inteira que representa o *makespan*

Função Objetivo:

$$\min C_{\max} \quad (5.1)$$

Sujeito a:

$$F_{i,K_{\max}} \leq C_{\max} \quad \forall i \quad (5.2)$$

$$\sum_t Z_{t,i,k} = 1 \quad \forall i, k \quad (5.3)$$

$$\sum_t X_{t,i,k} = \rho_{i,k} \quad \forall i, k \quad (5.4)$$

$$\sum_{t'=t}^{t+\rho_{i,k}} X_{t',i,k} \geq \rho_{i,k} Z_{t,i,k} \quad \forall i, k, t = 1, \dots, T - \rho_{i,k} \quad (5.5)$$

$$S_{i,k} = \sum_t t Z_{t,i,k} \quad \forall i, k \quad (5.6)$$

$$S_{i,k+1} = F_{i,k} \quad \forall i, k \quad (5.7)$$

$$F_{i,k} - S_{i,k} = \rho_{i,k} \quad \forall i, k \quad (5.8)$$

$$\sum_i \sum_k r_{i,p,k} X_{t,i,k} \leq C_p \quad \forall t, p \quad (5.9)$$

A função objetivo (5.1) minimiza o *makespan* dos trabalho, ou seja, o tempo entre o início do primeiro exame, e o fim do último.

A restrição (5.2) garante que todos os trabalho acabam antes ou no instante do *makespan*, auxiliando a função objetivo.

A restrição (5.3) garante que só existe um instante em que a operação k do trabalho i é atribuído.

A restrição (5.4) garante que para a operação k do trabalho i o somatório de $X_{t,i,k}$ sobre t tem o valor da duração $\rho_{i,k}$.

A restrição (5.5) garante a concordância do posicionamento sobre t entre $X_{t,i,k}$ e $Z_{t,i,k}$.

A restrição (5.6) garante a coerência do momento de início da tarefa k .

A restrição (5.7) garante que para o trabalho i a operação $k + 1$ começa quando a operação k acaba, verificando a restrição de *No-Wait*.

A restrição (5.8) garante que a duração da operação k para o trabalho i é de $\rho_{i,k}$.

A restrição (5.9) garante que não existe a sobre-utilização do recursos p durante o instante

t verificando as extensões *Multi-Resource* e *Flexible*.

Por outro lado, será possível formular modelos mais eficientes que o clássico. A formulação aqui sugerida, *MILP- δ* , tira partido do facto da restrição *No-Wait* criar um padrão de utilização de recursos, qualquer que seja o instante t onde este se insere. Desta forma não é necessário explicitamente formular restrições que garantem *No-Wait*.

Conjuntos:

O conjunto de trabalhos $I, i \in I := (1, \dots, n)$

O conjunto de recursos-tipo $R, p \in R := (1, \dots, R_{\max})$

O conjunto de instantes de tempo $T, t \in T := (1, \dots, T_{\max})$

Parâmetros:

ρ_i é duração do trabalho i

$\delta_i(u, p)$ é a quantidade de recursos do tipo p necessários a um offset de u instantes de tempo no trabalho i

C_p é a capacidade do recursos do tipo p

Variáveis de Decisão:

$Z_{t,i}$ é uma variável binária com valor 1 se o trabalho i começar no instante t , caso contrário tem valor 0

F_i é uma variável inteira que representa o valor de fim do trabalho i

C_{\max} é uma variável inteira que representa o *makespan*

Função Objetivo:

$$\min C_{\max} \quad (5.10)$$

Sujeito a:

$$F_i \leq C_{\max} \quad \forall i \quad (5.11)$$

$$\sum_{t=0}^{T_{\max}-\rho_i+1} Z_{t,i} = 1 \quad \forall i \quad (5.12)$$

$$\sum_{t=0}^{T_{\max}-\rho_i+1} Z_{t,i} * (t + \rho_i) = F_i \quad \forall i \quad (5.13)$$

$$\sum_i \sum_{\tau=\max(0, t-\rho_i+1)}^{\min(t, T_{\max}-\rho_i)} \delta_i(t - \tau, p) Z_{\tau,i} \leq C_p \quad \forall t, p \quad (5.14)$$

A função objetivo (5.10) minimiza o *makespan* dos trabalho.

A restrição (5.11) garante que todos os exames acabam antes ou no instante do *makespan*, auxiliando a função objetivo.

A restrição (5.12) garante que só existe um instante de começo do trabalho i .

A restrição (5.13) faz a ligação entre as variáveis $Z_{t,i}$ e F_i .

A restrição (5.14) garante que não há sobre utilização de recursos, ao verificar para cada instante t quais trabalhos estão a ser executados e em que fase este se encontra, garantindo que em cada instante não há sobre-utilização de recursos.

Para ambas das formulações apresentadas, o tamanho do conjunto de instantes de tempo, T , tem impacto no número de variáveis e restrições, será por isso importante encontrar o menor valor de T_{\max} possível mas que seja maior que *makespan*, a isto tipicamente chama-se de *upper-bound*.

De seguida foram desenvolvidos três modelos com duas codificações de solução diferentes. Os dois primeiros modelos utilizam uma codificação que representa o instante de início de cada trabalho. O outro modelo utiliza a sequencia de escalonamento de cada trabalho como solução.

5.1.1 Modelo 1

Este modelo é definido pelo facto de não existir a garantia da viabilidade de cada solução. Por isso será necessário criar uma função objetivo diferente das restantes, esta será dada por:

$$f(s) = C_{\max} + P \sum_{t=start}^{end-1} \sum_p v(t, p)$$

Onde C_{\max} é *makespan*, P a punição relativamente à sobre-utilização de recursos (sendo esta uma variável a estudar), esta sobre-utilização é dada pelo somatório sobre os recursos e instantes de tempo, $v(t, p)$ por sua vez é dado por:

$$v(t, p) = \begin{cases} 1 & \text{se } u(t, p) > C_p \\ 0 & \text{caso contrário} \end{cases}$$

Temos que $u(t, p)$ é a utilização de cada recurso p em cada instante t , C_p será a capacidade máxima de cada recurso. Poderia ter-se considerado uma definição diferente de $v(t, p)$.

$$v(t, p) = \begin{cases} u(t, p) - C_p & \text{se } u(t, p) > C_p \\ 0 & \text{caso contrário} \end{cases}$$

Apesar de corresponder melhor à verdadeira sobre-utilização de recursos, ao diferenciar a quantidade, esta formulação apresenta piores resultados.

Será de saliente como C_{\max} é calculado, existem duas formas semelhantes para tal. A primeira considera que C_{\max} é dado pelo valor máximo do término de cada trabalho. A

segunda será pela diferença entre o valor máximo de término de cada trabalho e o valor mínimo de início de cada trabalho. Apesar de aparentemente iguais, a diferença deste cálculo terá impacto na procura e na avaliação de cada vizinho.

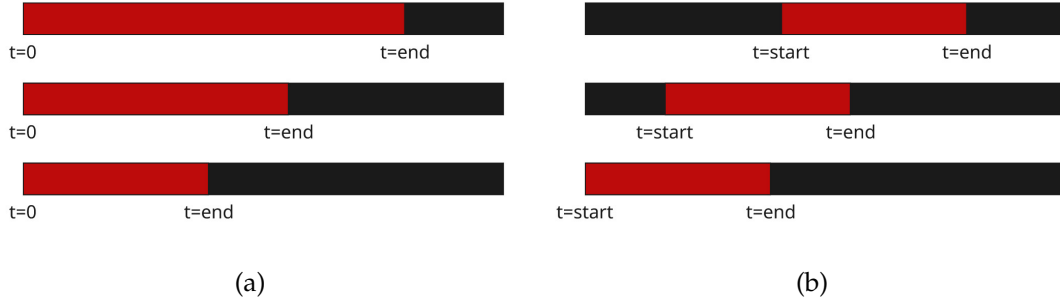


Figura 5.1: Impacto das diferentes formas de calcular C_{\max} .

A figura 5.1 apresenta três agendas diferentes, a vermelho temos o valor de *makespan* de cada agenda, na figura 5.1a temos que *makespan* é dado pelo máximo do término de cada trabalho, enquanto na figura 5.1b temos que *makespan* é dado pela diferença entre o valor máximo de término de cada trabalho e o valor mínimo de início de cada trabalho. Nesta primeira figura as três agendas têm valores diferentes de $f(s)$, enquanto que a segunda figura apresenta três agendas com o mesmo valor de $f(s)$.

Para este modelo, a solução s é dada pelo instante t em que começa cada trabalho i . Por sua vez, a vizinhança de s , $\mathcal{N}(s)$, é definida pelo trabalho a reagendar e quando este deve começar, (i, t) . Ambos estes valores são obtidos aleatoriamente, contudo t poderá ser obtido através de uma uniforme $U(0, \text{makespan} - \text{duração})$, ou $U(\text{ini}, \text{fim} - \text{duração})$, sendo a duração o tempo necessário para completar o trabalho i . Esta diferença aparentemente pequena gera soluções finais bastante diferentes.

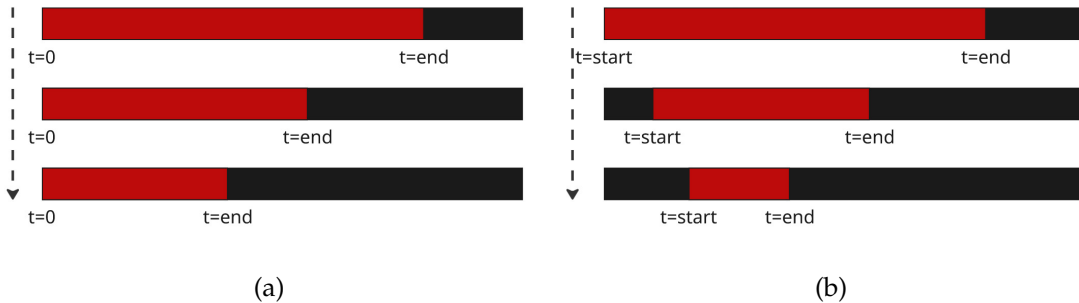


Figura 5.2: Impacto das duas forma de gerar t sobre a procura da vizinhança.

Podemos utiliza a figura 5.2 para perceber o porquê. Observamos uma sequencia no tempo da otimização de uma agenda, em que na figura 5.2a se utiliza $U(0, \text{makespan} - \text{duração})$ na geração de novos vizinhos, e na figura 5.2b se utiliza $U(\text{start}, \text{end})$ na geração

de novos vizinhos. Em ambas existe a diminuição de *makespan* quando se agenda o último trabalho para um instante inferior. Contudo apenas utilizando $U(\text{start}, \text{end-duração})$ para gerar os instantes poderá ocorrer a diminuição de *makespan* através do valor final e do inicial.



Figura 5.3: Processo de transição da solução s para a solução s' para o modelo 1.

O processo de modificar a solução atual s de forma a obter uma nova solução s' da vizinhança é resumido à figura 5.3. Sendo depois necessário calcular $f(s')$. O pseudo-código abaixo descreve este processo.

Devo incluir este algoritmo? Se sim, terá de ser muito mais reduzido como o expresso.

Algorithm 1: Pseudo-código de gerção de novos vizinhos, a sua avaliação, aceitação ou rejeição. Modelo 1.

```
 $i \leftarrow U(0, n);$ 
 $q \leftarrow U(0, 1);$ 
 $\delta \leftarrow \text{tipo\_exame\_delta}[i];$ 
//  $\delta$  é uma matriz com os recursos ao longo da duração do trabalho  $i$ 
 $d \leftarrow \text{duração}[i];$ 
 $\text{ini\_velho} \leftarrow S[i, 0];$ 
 $\text{ini\_novo} \leftarrow U(\text{ini}, \text{fim}-d);$ 
// ini e fim provenientes da iteração anterior
 $\text{salto} \leftarrow \text{ini\_novo}-\text{ini\_velho};$ 
for  $k \leftarrow 0$  to  $n_{\text{operações}}$  do
     $S[i, k] \leftarrow S[i, k] + \text{salto};$ 
     $F[i, k] \leftarrow F[i, k] + \text{salto};$ 
    //  $S$  e  $F$  são matrizes com o instante  $t$  de início e fim de cada
    trabalho  $i$  e operação  $k$ 
end
for  $t \leftarrow 0$  to  $d$  do
    for  $p \leftarrow 0$  to  $n_{\text{recursos}}$  do
         $u[t + \text{ini\_velho}, p] \leftarrow u[t + \text{ini\_velho}, p] - \delta[t, p];$ 
         $u[t + \text{ini\_novo}, p] \leftarrow u[t + \text{ini\_novo}, p] + \delta[t, p];$ 
    end
end
 $\text{ini} \leftarrow \min(S);$ 
 $\text{fim} \leftarrow \max(F);$ 
 $C_{\max} \leftarrow \text{fim} - \text{ini};$ 
 $f(s') \leftarrow C_{\max} + P \sum_{t=\text{start}}^{\text{end}-1} \sum_p v(t, p);$ 
if  $f(s') < f(s)$  ou  $q < \exp(-(f(s) - f(s'))/c)$  then
    |  $s \leftarrow s'$ 
end
else
    for  $k \leftarrow 0$  to  $n_{\text{operações}}$  do
         $S[i, k] \leftarrow S[i, k] - \text{salto};$ 
         $F[i, k] \leftarrow F[i, k] - \text{salto};$ 
    end
    for  $t \leftarrow 0$  to  $d$  do
        for  $p \leftarrow 1$  to  $n_{\text{recursos}}$  do
             $u[t + \text{ini\_velho}, p] \leftarrow u[t + \text{ini\_velho}, p] + \delta[t, p];$ 
             $u[t + \text{ini\_novo}, p] \leftarrow u[t + \text{ini\_novo}, p] - \delta[t, p];$ 
        end
    end
end
```

A temperatura inicial segue o esquema **IT6 franzinRevisitingSimulatedAnnealing2019**. Mais especificamente, gerou-se uma boa solução inicial utilizando a heurística NEH, e de seguida procedeu-se a uma caminhada aleatório da vizinhança desta solução utilizando $c = \infty$ e $L_k = 100000$, de forma a aceitar todas as novas soluções encontradas. Definindo-se então que a temperatura inicial é dada por:

$$c_0 = |\Delta_{avg} / \log(p_0)|$$

Onde Δ_{avg} é a diferença média entre a solução atual s e o vizinho s' , e p_0 é a probabilidade inicial de aceitar cada solução, este último valor será uma das variáveis a estudar.

O critério de paragem, CP , utilizado foi uma modificação de **SC9 franzinRevisitingSimulatedAnnealing2019**, onde se termina o algoritmo quando um número fixo de reduções de temperatura não geram novas soluções melhores, este valor será uma das variáveis a estudar.

O critério de aceitação utilizado foi o tradicional descrito por Metropolis [12], ou seja, **AC1 franzinRevisitingSimulatedAnnealing2019**.

O esquema de arrefecimento utilizado foi **CS2 franzinRevisitingSimulatedAnnealing2019**, dado por $c_{k+1} = \alpha c_k$, α tipicamente é um valor perto de 1, de forma a que a redução da temperatura se dê lentamente, este valor será uma das variáveis a estudar.

O comprimento da temperatura, ou seja, L_k , será uma variável a estudar, adotou-se o critério **TL1 franzinRevisitingSimulatedAnnealing2019**, considerando um número fixo de iterações durante cada redução de temperatura.

Existem três diferentes algoritmos considerados para a geração da solução inicial, um algoritmo *greedy*, um algoritmo aleatório, e a heurística NEH. O primeiro agenda cada trabalho aleatoriamente de forma a que o fim do trabalho i corresponde ao início do trabalho $i + 1$, garantindo que esta solução não apresenta sobre-utilização de recursos. O segundo agenda cada trabalho de forma aleatória, sem atender à possível sobre-utilização de recursos. Verificou-se que o segundo algoritmo apresenta, na sua generalidade, melhores soluções em relação ao primeiro algoritmo.

Podemos observar a figura 5.4 de acordo com o comportamento de cada solução e o porquê de rejeitarmos a heurística NEH como solução inicial. Em todos os algoritmos observa-se um pico da função objetivo, este facto deve-se à elevada temperatura inicial, levando conseqüente à aceitação da maioria das soluções candidatas, permitindo haver alguma independência entre a solução inicial e a final. Com o decair da temperatura a aceitação contrai, havendo aqui grande parte da melhoria observada. Quando a temperatura é muito reduzida, *SA* transforma-se numa simples procura local, ao tornar-se quase impossível aceitar soluções piores que a atual.

Apesar da heurística NEH apresentar uma boa solução inicial, este facto torna-se prejudicial no decorrer do *SA*, porque dificilmente haverá uma solução melhor que a inicial nas primeiras reduções de temperatura, podendo nunca melhorar até que o critério de paragem seja acionado.

As temperaturas iniciais encontradas são muito elevadas, Δ_{avg} é elevado devido à grande

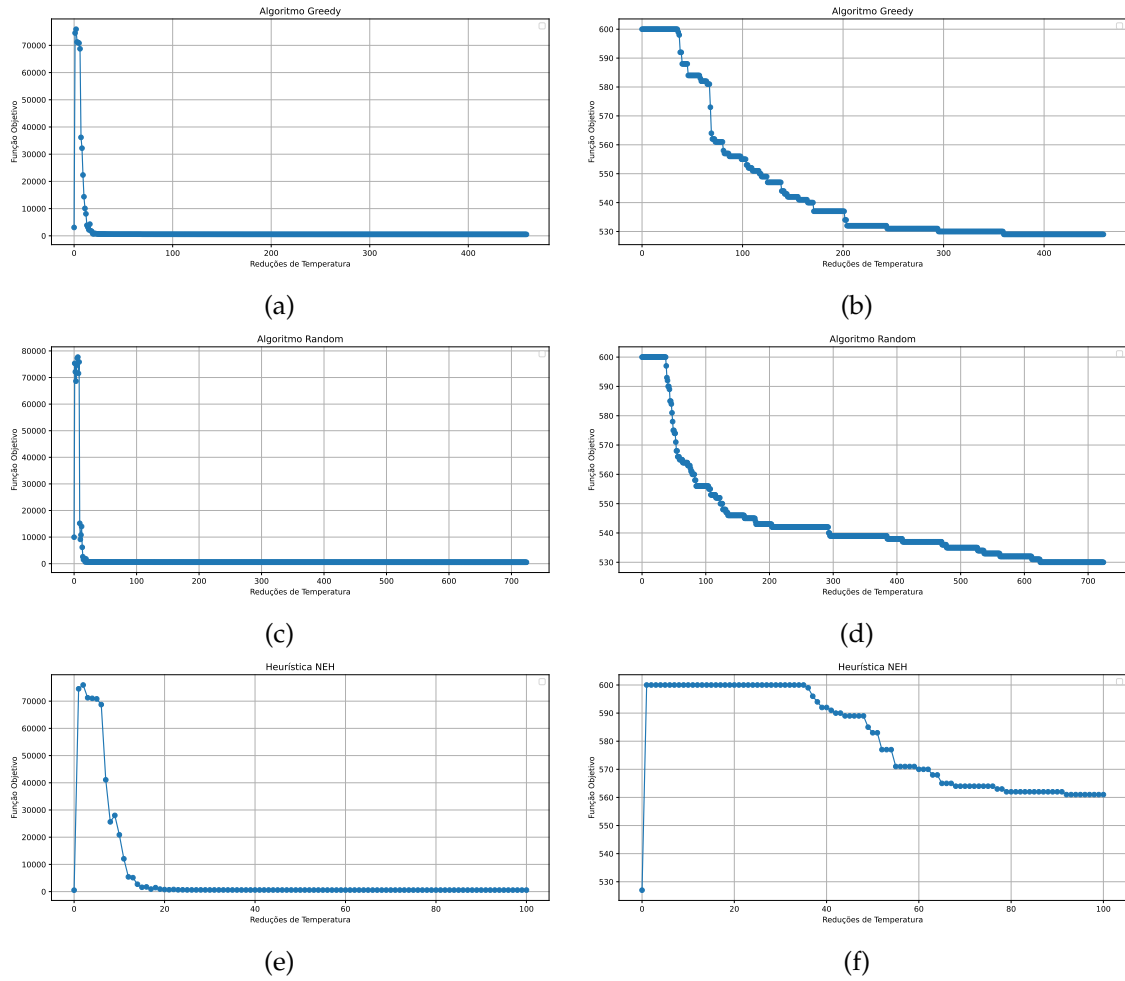


Figura 5.4: Evolução do valor da função objetivo ao longo da otimização do problema *makespan* com o modelo 1, para a obtenção destas figuras, as variáveis utilizadas foram: $L_k = 5000$, $CP = 100$, $\alpha = 0.8$, $P = 1000$, $p_0 = 0.9$.

probabilidade que cada solução gerada ter vários instantes com sobre-utilização de recursos, e devido ao elevado valor de P utilizado.

Será de realçar ainda duas peculiaridades deste modelo. Depois da redução da temperatura, o primeiro vizinho terá de ter o seu instante t gerado utilizando $U(0, T_{\max} - \text{duração})$, sendo T_{\max} o somatório das durações de todos os trabalhos, caso contrário o algoritmo parece não funcionar (não tenho explicação para isto).

Na rejeição de uma solução deveria-se recalculer o início e o fim da agenda, de forma a que, na próxima iteração, o novo instante gerado seja gerado com os valores de ini e fim corretamente à solução atual. Contudo isto resulta em piores soluções finais.

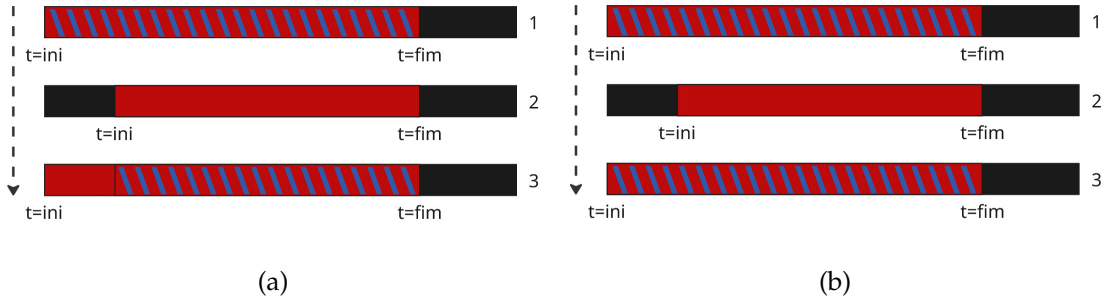


Figura 5.5: Impacto do recálculo de ini e fim sobre os instantes considerados na próxima iteração.

Na figura 5.5, a primeira agenda tem $makespan = fim - ini$, e por isso o vizinho terá um novo instante gerado por $U(ini, fim - duração)$, a segunda agenda mostra as fronteiras depois de se rejeitar a solução, a terceira agenda em 5.5a irá gerar o novo instante numa secção mais restrita da agenda, guiando $makespan$ para ser menor, por sua vez em 5.5b recalcula-se os valores de ini e fim, tal que toda a amplitude de $makespan$ é considerada é considerada na geração de um novo instante. Este é o único mecanismo de enviesamento da solução neste modelo.

5.1.2 Modelo 2

Este modelo é semelhante ao anterior, contudo existe garantia da viabilidade de cada solução durante a geração da vizinhança. Desta forma, a função objetivo é mais simples, e é dada por:

$$f(s) = C_{\max}$$

Sendo C_{\max} o valor de $makespan$ calculado como anteriormente, a diferença entre o máximo de término de cada trabalho e o valor mínimo de início de cada trabalho.

A codificação utilizada para este modelo é igual à do anterior. Contudo a forma de gerar a vizinhança, especificamente o novo instante t que um dado trabalho deve começar, difere substancialmente.

Como existe garantia da viabilidade da solução, ou seja, todas as restrições são obedecidas, não podemos seleccionar qualquer instante t para iniciar o trabalho i . Será preciso por isso um algoritmo que encontre todos os instantes onde se pode inserir o trabalho i sem ocorrer sobre-utilização dos recursos. Para tal recorreu-se ao seguinte algoritmo:

Algorithm 2: Pseudo-código que gera os candidatos durante a criação da vizinhança.

```

Candidatos = {};
for  $t \leftarrow ini$  to  $fim - d$  do
     $u[t : t + d] \leftarrow u[t : t + d] + \delta$ ;
    if  $u \leq C_p$  then
        Candidatos  $\leftarrow$  Candidatos  $\cup \{t\}$ ;           // Adicionar  $t$  à lista de
        candidatos
    end
end
Candidatos  $\leftarrow$  Candidatos  $\cup \{\max(ini - d, 0)\}$ ;
Candidatos  $\leftarrow$  Candidatos  $\cup \{\min(fim + 1, T_{max} - d)\}$ ;
return Candidatos

```

Verifica-se para cada instante t se é possível inserir o exame i sem ocorrer sobre-utilização de recursos, caso não haja, então o instante t é adicionado à lista de candidatos. A partir dessa lista, é selecionado aleatoriamente um instante t que será quando o trabalho i começa.

Foi necessário incluir na lista de candidatos um instante que se encontra-se fora da solução atual, antes do início com o valor $\max(ini - d, 0)$ ou depois do fim, com o valor $\min(fim + 1, T_{max} - d)$, caso contrário não se verificava nenhuma exploração de soluções, devido ao facto que os candidatos frequentemente eram apenas o mesmo instante da solução atual, causando uma temperatura inicial próxima de 0.

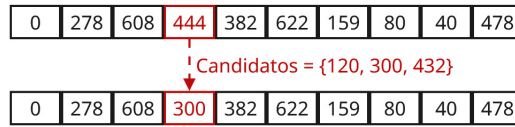


Figura 5.6: Processo de transição da solução s para a solução s' para o modelo 2.

O processo de modificar a solução s para uma solução s' resume-se à figura 5.6. Encontra-se os candidatos com o algoritmo 2 e depois escolhe-se aleatoriamente um para o instante de começo. O pseudo-código abaixo descreve este processo:

Algorithm 3: Pseudo-código de geração de novos vizinhos, a sua avaliação, aceitação ou rejeição. Modelo 2.

```

 $i \leftarrow U(0, n);$ 
 $q \leftarrow U(0, 1);$ 
 $d \leftarrow \text{duração}[i];$ 
 $t_{velho} \leftarrow S[i];$ 
// Retirar da matriz de utilização de recursos  $u$ , a utilização
  relativa a  $i$ 
 $u_{[t_{velho}:t_{velho}+d]} \leftarrow u_{[t_{velho}:t_{velho}+d]} - \delta;$ 
 $t_{novo} \leftarrow \text{Candidatos}[U(0, |\text{Candidatos}| - 1)];$ 
// Modificar a matriz  $S$  no índice  $i$  de forma a começar no novo
  instante  $t_{novo}$ 
 $S[i] \leftarrow t_{novo};$ 
// Adicionar à matriz de utilização de recursos  $u$ , a utilização
  relativa a  $i$ 
 $u_{[t_{novo}:t_{novo}+d]} \leftarrow u_{[t_{novo}:t_{novo}+d]} + \delta;$ 
 $ini \leftarrow \min(S);$ 
 $fim \leftarrow \max(S + \text{duração});$ 
 $C_{\max} \leftarrow fim - ini;$ 
 $f(s') \leftarrow C_{\max};$ 
if  $f(s') < f(s)$  ou  $q < \exp((f(s) - f(s'))/c)$  then
  |  $s \leftarrow s'$ 
end
else
  |  $S[i] \leftarrow t_{velho};$ 
  |  $u_{[t_{velho}:t_{velho}+d]} \leftarrow u_{[t_{velho}:t_{velho}+d]} + \delta;$ 
  |  $u_{[t_{novo}:t_{novo}+d]} \leftarrow u_{[t_{novo}:t_{novo}+d]} - \delta;$ 
end

```

Foram utilizados os mesmo critérios do modelo anterior, descritos em Franzin et al. **franzinRevisitingSimulatedAnnealing2019**, ou seja, **NE1**, **UT6**, uma modificação de **SC9**, **AC1**, **CS2**, e **TL1**.

Foram considerados novamente as três diferentes soluções iniciais, o algoritmo *greedy*, o algoritmo aleatório, e a heurística NEH. Podemos observar, na figura 5.7, como cada solução inicial se propaga ao longo do tempo de acordo com o *makespan*. Evidentemente a situação é semelhante à do modelo anterior, observa-se um pico inicial do valor da função objetivo representando a exploração originando independência relativamente à solução inicial. De seguida a temperatura é reduzida substancialmente, ocorrendo a caminhada aleatória que minimização o *makespan*. Finalmente, aciona-se o critério de paragem, verificando que não existe solução local

melhor. A temperatura inicial utilizada neste modelo é muito inferior ao anterior, Δ_{avg} será sempre inferior à duração do maior trabalho.

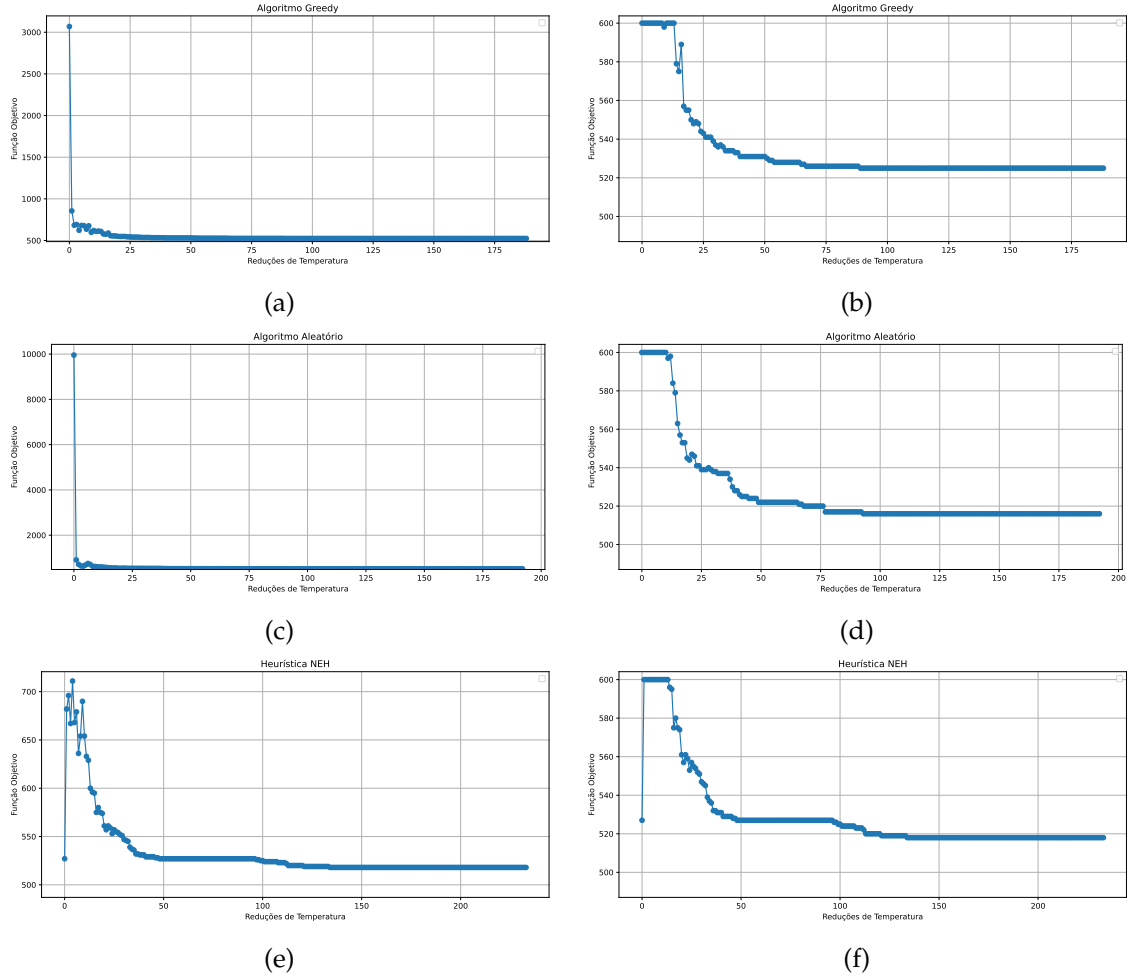


Figura 5.7: Evolução do valor da função objetivo ao longo da otimização do problema *makespan* com o modelo 2, para a obtenção destas figuras, as variáveis utilizadas foram: $L_k = 500$, $CP = 100$, $\alpha = 0.9$, $P = 0$, $p_0 = 0.9$.

De forma a manter a uniformidade, manteremos o algoritmo aleatório na geração de soluções iniciais. A solução inicial gerada pela heurística NEH continua a ser boa em relação ao modelo, por isso será de esperar que este não consiga encontrar nenhuma solução melhor antes de acionar CP prematuramente, descartamos por isso a heurística. Este modelo tem uma peculiaridade, como a temperatura inicial é reduzida, acabamos por verificar pouco impacto com a variação de α e p_0 , através da tabela .7 e .8. Este facto é agravado quando, ao rejeitar uma solução, recalcula-se os valores de ini e fim, o que irá causar que poucos instantes além da agenda inicial sejam considerados. Se não recalcularmos o início e o fim da agenda estarão mais instantes disponíveis para serem selecionados.

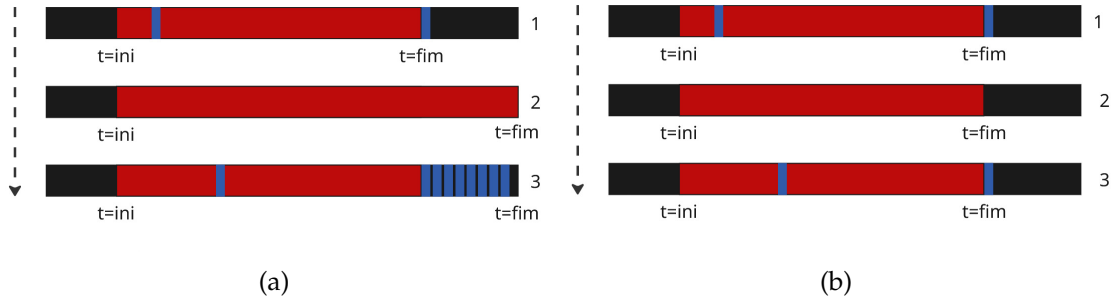


Figura 5.8: Impacto do recálculo de ini e fim sobre a lista de candidatos da próxima iteração.

Pela figura 5.8 realça-se a diferença discutida. Na figura 5.8a a primeira agenda mostra os candidatos existentes para a solução atual. Ao escolher o segundo candidato e rejeitarmos esta solução, obtemos a segunda agenda, que apresenta os valores de início e fim. A terceira mostra o número de candidatos disponíveis, existiram mais candidatos que, apesar de aumentarem o *makespan*, fazem-no de forma razoável, permitindo explorar mais intensivamente a vizinhança o que aumenta a probabilidade de sair de um mínimo local. Por sua vez, na figura 5.8b o número de candidatos disponíveis tende a ser muito menor, frequentemente sendo apenas uma escolha entre o instante atual, ou um instante que fortemente impacta *makespan*, tornando a exploração da vizinhança mais reduzida.

5.1.3 Modelo 3

Este modelo continua a garantir a viabilidade de cada solução considerada, por isso a função objetivo não será mais que o valor de *makespan*.

Contudo a codificação utilizada difere fortemente dos modelos anteriores. Em vez de representar uma solução pelo tempo de começo de cada trabalho, a solução é representada pela sequência pela qual se agenda cada trabalho, iremos representar esta sequência por π .

Esta sequência pouca informação nos fornece por si só, nomeadamente está ausente o instante em que cada trabalho deverá começar. Para tal é necessário um algoritmo auxiliar que descodifica a sequência de forma a obter a agenda, a este algoritmo chamamos de *timetabling*. Como já referido, existem vários algoritmos utilizados para este fim, contudo apenas dois vão ser comparados neste modelo, *left-shift timetabling* e *enhanced left-shift timetabling*.

Algorithm 4: Pseudo-código do algoritmo para encontrar o instante mais cedo possível.

```
for  $t \leftarrow ini$  to  $fim - d$  do
     $u[t : t + d] \leftarrow u[t : t + d] + \delta$ ;
    if  $u \leq C_p$  then
        return  $t$ 
    end
end
```

Algorithm 5: Pseudo-código de *left-shift timetabling*

```
Tempos  $\leftarrow \{\}$ ;
for  $i \leftarrow 0$  to  $|Sequência|$  do
     $d \leftarrow duração[i]$ ;
     $\delta \leftarrow tipo\_exame\_delta[i]$ ;
     $t \leftarrow instante\_mais\_cedo$ ;
    Tempos  $\leftarrow Tempos \cup \{t\}$ ;
     $u[t:t+d] \leftarrow u[t:t+d] + \delta$ ;
end
```

Algorithm 6: Pseudo-código de *enhanced left-shift timetabling*

```
Tempos  $\leftarrow \{\}$ ;
for  $i \in Sequência$  do
     $d \leftarrow duração[i]$ ;
     $\delta \leftarrow tipo\_exame\_delta[i]$ ;
     $t \leftarrow instante\_mais\_cedo$ ;
    Tempos  $\leftarrow Tempos \cup \{t\}$ ;
     $u[t:t+d] \leftarrow u[t:t+d] + \delta$ ;
end
Sequência  $\leftarrow Sequência.flip()$ ;
for  $i \in Sequência$  do
     $d \leftarrow duração[i]$ ;
     $\delta \leftarrow tipo\_exame\_delta[i]$ ;
     $\delta \leftarrow \delta.flip()$ ;
     $t \leftarrow instante\_mais\_cedo$ ;
    Tempos  $\leftarrow Tempos \cup \{t\}$ ;
     $u[t:t+d] \leftarrow u[t:t+d] + \delta$ ;
end
```

O algoritmo 4 permite encontrar o instante de tempo t mais cedo onde é possível agendar o trabalho i que se está a agendar. A forma de como isto é feito é algo bruto, pois percorre-se cada instante t e insere-se o trabalho i verificando se existe, ou não, sobre-utilização de recursos, este algoritmo consome a maioria do tempo computacional

deste modelo.

Uma adaptação pode ser utilizada, caso se conheça quais os recursos limitantes, ou seja, aqueles que mais provavelmente originaram sobre-utilização, pode-se primeiro verificar apenas esses recursos, e só depois verificar os restantes. Esta adaptação foi implementada, utilizando como recursos limitantes SCG e TM.

O algoritmo de *timetabling* 5 encontra, para cada elemento da solução, o instante de tempo mais cedo onde este pode ser inserido, modificando a matriz de utilização de recursos. Na literatura admite-se que assegurar que $t_{\pi[i]} \geq t_{\pi[i-1]}$, ou seja que cada elemento da sequência deve começar no mesmo instante ou depois do elemento anterior, tende a reduzir a qualidade do algoritmo, por isso esta característica não é assegurada. O algoritmo de *timetabling* 6 é semelhante, para cada elemento da sequência $\pi = \pi[0], \dots, \pi[n-1]$, encontra o instante mais cedo onde cada elemento pode ser inserido. Contudo, faz o mesmo processo para a sequência inversa, $\pi = \pi[n-1], \dots, \pi[0]$, em que cada trabalho também é o inverso, ao inverter δ . Não há a garantia que um ou outro algoritmo é melhor, mas existe a possibilidade de serem geradas soluções melhores com este passo adicional. Em contrapartida, o tempo computacional associado à descodificação da solução duplica.

Para demonstrar a diferença entre estes dois algoritmos irá utilizar-se o exemplo apresentado por Ying & Lin [28], onde é apresentado um pequeno problema *No-Wait Job-Shop* definido por:

Tabela 5.1: Problema definido pela operações de cada trabalho e os recursos necessários e tempo de processamento, apresentado por Ying & Lin [28]

Trabalho	Sequência de recursos (tempo)			
J_1	4(7)→	3(9)→	1(8)→	2(4)
J_2	1(4)→	4(8)→	2(2)→	3(4)
J_3	3(5)→	2(7)→	1(1)→	4(4)
J_4	4(3)→	2(1)→	1(2)→	3(3)

Para cada um dos quatro trabalhos deste problema, a tabela 5.1 apresenta a sequência das operações que devem ser realizadas, e o respetivo tempo de processamento. Ou seja, o trabalho J_1 primeiro passa pelo recurso 4 com tempo de processamento de 7 unidades de tempo, a segunda operação será no recurso 3 durante 9 unidades de tempo, a terceira será no recurso 1 durante 8 unidades de tempo, a quarta operação será no recurso 2 durante 4 unidades de tempo.

O exemplo dado utiliza a sequência 3-4-1-2, ou seja, agenda-se o mais cedo possível o trabalho 3, depois o trabalho 4, de forma consecutiva sem garantir que $t_3 \leq t_4 \leq t_1 \leq t_2$. Podemos ver, na figura 5.9 a agenda *forward*, e a agenda *backward* com a sequência inversa e com cada trabalho também invertido, é evidente que das duas agendas apresentadas iremos escolher aquela que apresentar o menor *makespan*, sendo neste caso a agenda

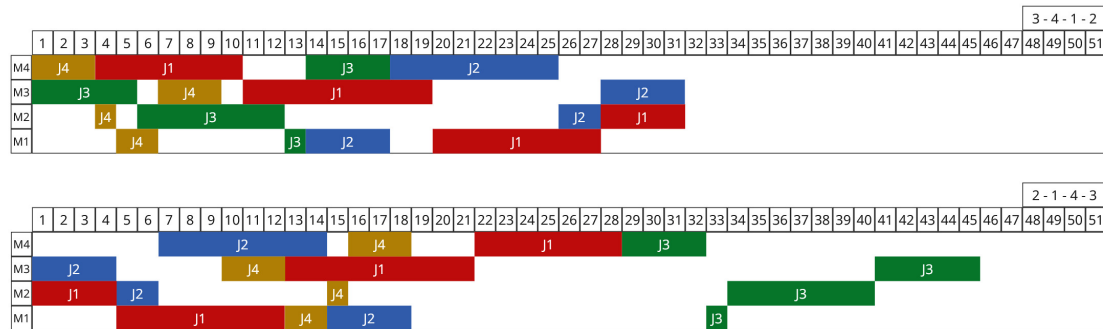


Figura 5.9: Exemplo da solução 3-4-1-2

forward.

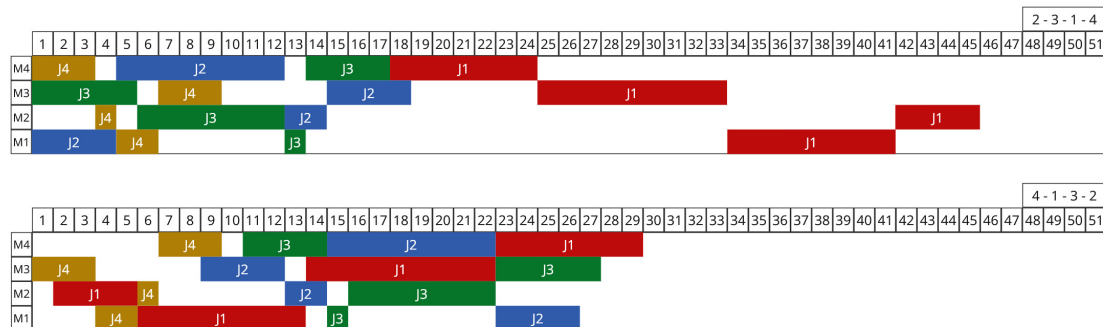


Figura 5.10: Exemplo da solução 2-3-1-4

Novamente, observamos outro exemplo de uma solução na figura 5.10, agora com a sequência 2-3-1-4, e duas agendas, uma *forward* e a outra *backward*, respectivamente. Este exemplo serve para demonstrar a utilidade de representar a mesma solução com mais de um algoritmo de *timetabling*. Podendo transformar uma solução aparente má numa boa solução.

Por sua vez, a vizinhança de uma solução pode ser gerada de três formas, com várias variantes. Através de uma troca de índice entre trabalhos, figura 5.11a. Através da retirada de um trabalho e a sua inserção noutra índice da sequência, figura 5.11b. Através da inversão de uma sub-sequência de trabalhos, figura 5.11c. Não existe, contudo nenhuma evidência que um dos métodos é superior ao outro, na continuação iremos utilizar a troca de trabalhos. Serão reportados os valores no apêndice asdasdasd asdasdada.

Foram utilizados os mesmo critérios do modelo anterior, descritos em Franzin et al. *franzinRevisitingSimulatedAnnealing2019*, ou seja, NE1, UT6, uma modificação de SC9, AC1, CS2, e TL1.

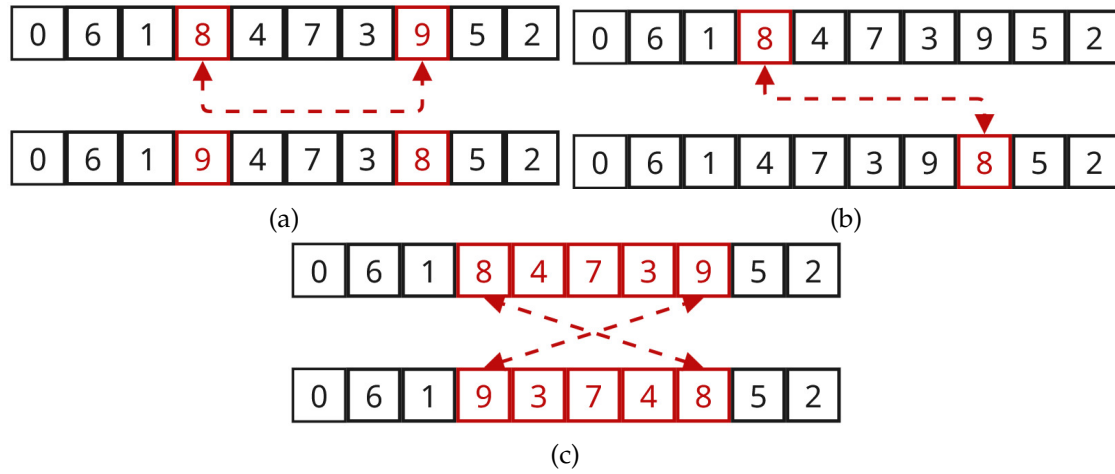


Figura 5.11: Processo de transição da solução s para a solução s' para o modelo 3.

A geração da solução inicial é diferente aos modelos anterior, devido à codificação diferente. A forma mais simples de gerar uma solução será pela escolha aleatória de uma sequência, existindo sempre garantia da viabilidade da solução graças ao algoritmo de *timetabling*. Também será possível utilizar a heurística de NEH para este fim.

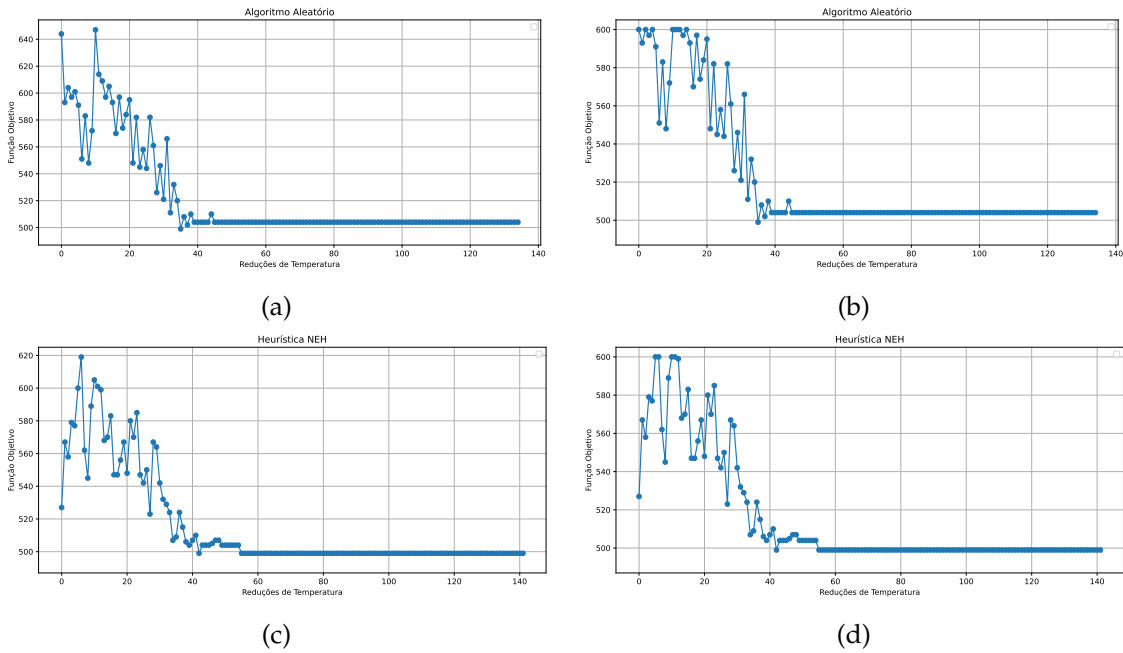


Figura 5.12: Evolução do valor da função objetivo ao longo da otimização do problema *makespan* com o modelo 3, para a obtenção destas figuras, as variáveis utilizadas foram: $L_k = 500$, $CP = 100$, $\alpha = 0.9$, $P = 0$, $p_0 = 0.9$, *left-shift timetabling*, geração de vizinhos através de troca.

Na figura 5.12 observamos como evolui *makespan* ao longo do processo de otimização. Ao contrário dos problemas anteriores, não existe inicialmente um pico tão extremo da função objetivo como nos modelos anteriores, devido ao baixo valor de Δ_{abg} . Este facto

será devido ao algoritmo de *timetabling* utilizado apresentar sempre uma solução perto de ótima para cada sequência.

Este modelo acaba por ser dividido em dois problemas distintos, o problema de sequenciamento e o de *timetabling* já discutido. Sem um bom algoritmo de *timetabling* não será possível obter boas soluções, mas a sequência é o que realmente define a solução.

RESULTADOS

Neste capítulo serão discutidos os resultados obtidos com a implementação dos vários modelos. Serão apresentados, para cada modelo, o impacto que as variáveis têm nas soluções em relação à qualidade e ao tempo computacional utilizado. Também serão utilizadas instâncias de *benchmark* da literatura.

6.1 Problema de *Makespan*

Para os três modelos já apresentados existem cinco variáveis comuns com impacto na qualidade e no tempo computacional de cada solução. Desta forma, será vital encontrar uma combinação de níveis que permite obter soluções boas de forma rápida.

Estas cinco variáveis são: L_k , o número de iterações, ou vizinhos visitados, entre arrefecimentos; CP , o critério de paragem, que representa quando deve ocorrer a paragem do algoritmo; α , a função de arrefecimento; P a punição relativa à sobre-utilização de recursos; p_0 , descrito como a probabilidade inicial de aceitar uma solução vizinha, algo que influencia a temperatura inicial.

Idealmente, utilizar-se-ia $L_k = \infty$, $CP = \infty$, $1 > \alpha \simeq 1$, $p_0 = 1$. Contudo isto implicaria um tempo computacional infinito, por isso irão ser procurados outros níveis para as variáveis.

Tabela 6.1: Níveis das variáveis dos modelos de *Simulated Annealing*

Modelo	L_k	CP	α	P	p_0
M1	500, 1500, 2500	10, 55, 100	0.8, 0.9, 0.975	10, 55, 100	0.5, 0.7, 0.9
M2	50, 150, 250	10, 55, 100	0.8, 0.9, 0.975	0	0.5, 0.7, 0.9
M3	50, 150, 250	5, 30, 55	0.8, 0.9, 0.975	0	0.5, 0.7, 0.9

Os níveis apresentados na tabela 6.1 foram escolhidos de forma a existir exploração da interação entre os fatores. Por outro lado, a diferença observada nos fatores CP e L_k deve-se ao facto destes serem os principais decisores do tempo computacional, a sua alteração entre modelos permitiu que o tempo computacional permanecesse comparável.

Para os próximos resultados, serão considerados os exames que decorrem numa segunda-feira habitual. Ou seja, 3 cintigrafia tiroideia, 5 cintigrafia pulmonar de ventilação/inalação + perfusão, 10 cintigrafia miocárdica de perfusão em repouso, 1 cintigrafia das glândulas salivares, 10 PET - estudo corpo inteiro com FDG.

Os resultados obtidos são provenientes de 100 repetições de cada combinação de variáveis. Como foi possível computar paralelamente 10 repetições, os valores apresentados são a média do valor mínimo de 10 repetições. Semelhantemente, o tempo computacional aprestado é relativo à média do valor máximo de 10 repetições.

Todos os valores foram obtidos utilizando um processador Ryzen 5600, os modelos SA foram executados em Python 3.11 e os modelos MILP em Gurobi 12.0.

6.1.1 Modelo 1

Será agora apresentada um diagrama de calor referente ao valor da função objetivo e tempo computacional. Para a função objetivo, a faixa de cores apresentada encontra-se entre 487 e 527, sendo estes os valores da melhor solução encontrada através do MILP e da heurística NEH, respetivamente. O tempo computacional, por sua vez teve a sua faixa de cores limitada entre 0 e 60 segundos.

Existem algumas observações que se podem fazer com a análise da figura 6.1, em relação ao impacto que cada variável tem.

Observa-se a melhoria dos resultados e o aumento do tempo computacional com o aumento de CP e L_k . Demonstrando novamente que estes são os principais decisores da qualidade e tempo computacional das soluções.

A interação entre valores altos de temperatura inicial, devido a p_0 alto, redução lenta da temperatura, devido a α alto, e valores baixos de CP , provocam muita aceitação inicial de soluções, levando a más soluções que não são melhoradas devido à permanência alta de temperaturas e rápida paragem do algoritmo.

O aumento de CP garante que se encontra uma solução igual ou melhor quando comparado a um nível de CP menor, isto ocorre porque CP não altera o percurso feito pelo algoritmo. O impacto do fator de punição P é difícil de quantificar. Intuitivamente, quando a punição é pequena, irá ocorrer mais exploração devido ao impacto reduzido sobre a função objetivo, por outro lado, quando a punição é grande, a exploração é menor mas há mais certeza que as soluções geradas não apresentarão sobre-utilização de recursos.

Por sua vez, o tempo computacional está correlacionado com a qualidade da solução. Uma combinação de níveis que apresentem melhores soluções vão reiniciar a critério de paragem, o que leva a um maior tempo computacional, apresentando correlação de Spearman de -0.92, quando a função objetivo diminui, o tempo computacional tende a aumentar.

Com a ajuda da tabela .5 do apêndice (.1) podemos tecer alguns comentários sobre os

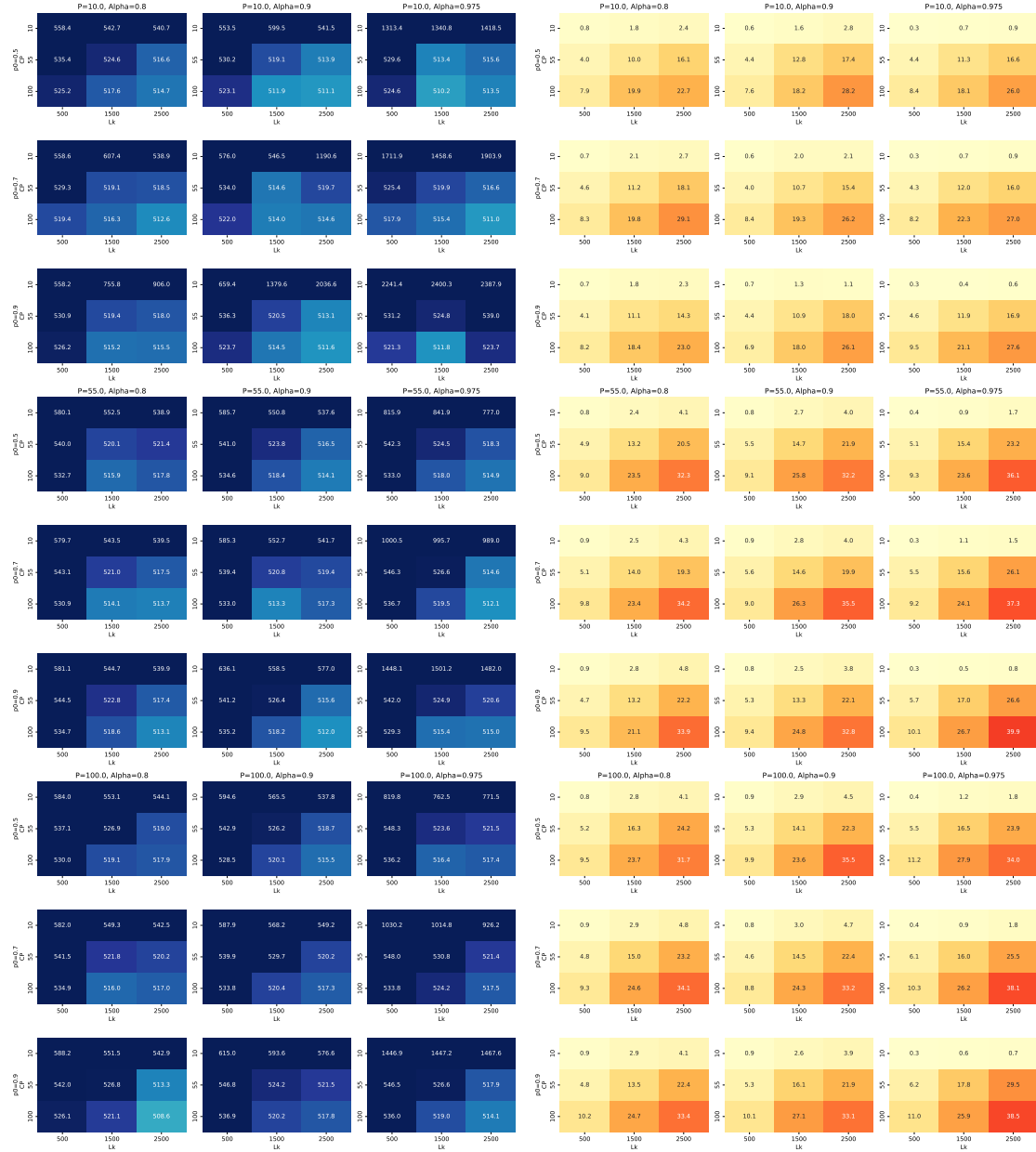


Figura 6.1: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 1.

valores ótimos das variáveis primárias, em relação ao *makespan*:

- L_k não apresenta efeito linear ou quadrático significativo;
- CP apresenta efeito linear e quadrático significativo. Intuitivamente sabemos que o melhor valor será o maior possível;
- α apresenta efeitos lineares e quadráticos significativos. Tendo em conta os efeitos de 250 e -90 respetivamente, podemos concluir que o valor deve ser o menor possível;

- P apresenta efeito linear e quadrático significativo. Considerando os efeitos de -120 e -60 respetivamente, o valor a utilizar será o maior valor possível;
- p_0 só apresenta efeito linear significativo, sendo o seu efeito positivo, devendo-se escolher o menor valor possível.

Relativamente ao tempo, poderemos fazer uma análise semelhante pela tabela .6:

- L_k apresenta efeitos lineares e quadráticos significativos, ambos com efeitos positivos, intuitivamente o valor que reduz o tempo computacional será o menor possível de L_k ;
- CP apresenta efeito linear e quadrático significativo, ambos com efeito positivo, outra vez intuitivamente o valor que reduz o tempo computacional será o menor valor possível de CP ;
- α tem efeito linear significativo, mas não quadrático. Através do valor do positivo do efeito, sabemos que idealmente será o menor valor possível;
- P apresenta efeito linear e quadrático significativo, ambos com efeito positivo, será difícil definir que nível minimiza o tempo computacional. Pelo fator linear escolheríamos P baixo, mas o efeito quadrático indica que o valor que minimiza não está nas extremidades;
- p_0 não tem efeito linear ou quadrático significativo. Escolheremos o valor que for mais conveniente.

De forma a obter uma boa combinação de variáveis, optou-se por definir uma função com ambas as variáveis dependentes a minimizar. A função é dada por:

$$makespan^{10} \times tempo^{0.1}$$

Esta fórmula foi escolhida porque pretendemos minimizar ambas as parcelas, *makespan* e o tempo, mas também pretendemos dar mais importância a pequenas diferenças entre valores de *makespan*. Desta forma, a combinação que minimiza esta fórmula é {1500, 100, 0.975, 10, 0.5} com *makespan* médio de 510.2 e tempo computacional médio de 18.1 segundos.

6.1.2 Modelo 2

Será apresentado um novo diagrama de calor referente ao *makespan* e ao correspondente tempo computacional obtido através do modelo 2 utilizando as várias combinações de níveis apresentadas na tabela 6.1. Outra vez, a faixa de cores estarão limitas aos valores já referidos.

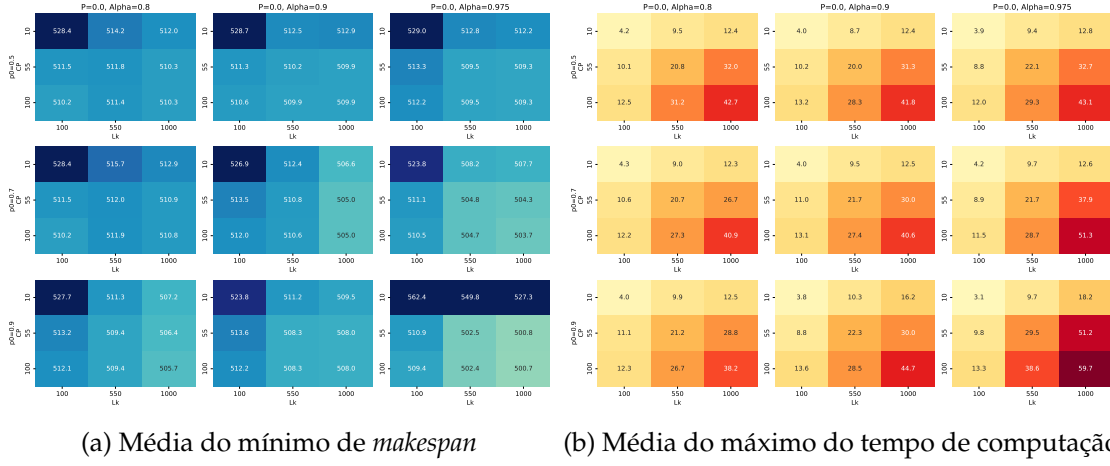


Figura 6.2: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 2.

Com a análise da figura 6.2 podemos tirar algumas conclusões acerca do impacto que cada variável tem sobre a qualidade da solução e o tempo computacional necessário. Observa-se de imediato que as soluções apresentadas são melhores do que no modelo 1, sem aumentos desproporcionais do tempo computacional, este facto é especialmente evidente por não existirem combinações de variáveis muito superiores a 527 como foi observado no modelo 1.

Observa-se novamente o declínio da qualidade das soluções quando CP é baixo, e p_0 e α são altos. Semelhantemente observa-se que ao aumentar CP garantidamente a solução mantém-se ou melhora, o mesmo não se verifica para L_k contudo é razoável admitir que existe a tendência para a solução melhorar.

Com a ajuda da tabela .7 podemos comentar como se comportam as variáveis dependentes de acordo com as variáveis independentes. Relativamente ao *makespan* temos que:

- L_k apresenta efeito linear significativo, mas não efeito quadrático significativo. Assim, será de esperar que o melhor valor de L_k seja o máximo que possamos utilizar;
- CP apresenta efeito linear e quadrático significativo. Intuitivamente também será de esperar que o melhor valor a utilizar seja o máximo;
- α e p_0 não apresentam efeito significativo, deveremos escolher o nível que for mais conveniente.

Relativamente ao tempo, poderemos fazer uma análise semelhante pela tabela .8:

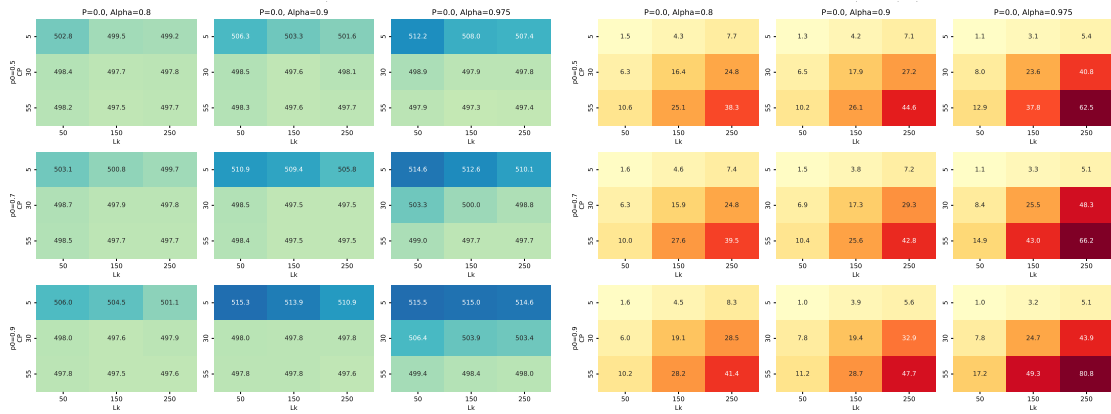
- L_k continua a ter efeito linear significativo mas não quadrático significativo, será de esperar que o melhor valor de L_k seja baixo, devido ao efeito positivo que apresenta;
- CP continua a ter efeito linear e quadrático significativo, mas experimentalmente sabemos que quanto menor for CP mais rápido termina a procura;

- α tem efeito linear significativo, mas não quadrático. Através do valor positivo do efeito sabemos que o idealmente será o menor valor possível;
- p_0 também tem efeito linear significativo, mas não quadrático. Outra vez, pelo valor positivo do efeito sabemos que o ideal será minimizar esta variável.

Desta forma, a combinação que minimiza a fórmula já descrita é $\{1000, 10, 0.9, 0, 0.7\}$, com *makespan* médio de 506.6 e tempo computacional médio de 12.5 segundos.

6.1.3 Modelo 3

Será outra vez apresentado um diagrama de calor referente ao *makespan* e ao tempo computacional obtido pelo modelo 3, com os níveis descritos na tabela 6.1. A faixa de cores continuam a ser limitadas como de antes.



(a) Média do mínimo de *makespan*

(b) Média do máximo do tempo de computação

Figura 6.3: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 3 com *left shifting*.

Na figura 6.3 observamos de imediato que o modelo 3 é melhor que os anteriores, apresentando soluções melhor com tempo computacional semelhante. Voltamos a retirar as mesmas observações, aumentar CP garantidamente mantém ou melhora a solução, quando CP é baixo, e p_0 e α são altos observamos soluções de pior qualidade. Podemos analisar qual o efeito primário de cada variável sobre o *makespan* e tempo, utilizando respetivamente a tabela 9 e tabela 10:

- L_k tem efeito linear significativo sobre o *makespan* e sobre o tempo, o mesmo não se verificando no efeito quadrático. Pelo valor do efeito, temos que deva ser o maior possível e o menor possível, respetivamente;
- CP tem efeito linear e quadrático significativo sobre o *makespan* e sobre o tempo. Experimentalmente sabemos que ao aumentar CP a solução mantém-se ou melhora mas consequentemente aumenta o tempo computacional;

- Para o *makespan*, α apenas apresenta efeito linear significativo, e de acordo com o efeito positivo, deveremos minimizar este valor. Por sua vez, α apresenta sobre o tempo efeito linear e quadrático significativo, com efeitos positivo e negativos respetivamente, desta forma deveremos utilizar um valor baixo de α para minimizar o tempo computacional;
- p_0 tem efeito linear significativo sobre o *makespan* e sobre o tempo, sendo o valor do efeito positivo em ambos, indicando que a concordância entre ambas as variáveis dependentes, devendo usar o menor valor possível.

Desta forma, a combinação que minimiza a fórmula já descrita é $\{50, 5, 0.8, 0, 0.5\}$, com *makespan* médio de 502.8 e tempo computacional médio de 1.5 segundos.

6.1.4 Comparação entre modelos

De seguida são apresentados os exames realizados em cada dia da semana no departamento de medicina nuclear em estudo. Para cada um destes dias, serão utilizados os modelos já descritos para encontrar uma solução que permita minimizar o *makespan*, cada modelo utilizará a combinação de variáveis já descritas.

- Segunda-feira: 3 cintigrafia tiroideia, 5 cintigrafia pulmonar de ventilação/inalação + perfusão, 10 cintigrafia miocárdica de perfusão em repouso, 1 cintigrafia das glândulas salivares, 10 PET - estudo corpo inteiro com FDG;
- Terça-feira: 2 cintigrafia tiroideia, 10 cintigrafia óssea corpo inteiro, 1 cintigrafia para amiloidose cardíaca, 1 linfocintigrafia para detecção de ganglio sentinela, 2 cintigrafia pulmonar de ventilação/inalação + perfusão, 10 PET - estudo corpo inteiro com FDG;
- Quarta-feira: 2 cintigrafia tiroideia, 2 linfocintigrafia para detecção de ganglio sentinela, 6 cintigrafia das paratiroideias, 1 tomografia cerebral com 123I-ioflupano, 3 cintigrafia pulmonar de ventilação/inalação + perfusão, 8 PET - estudo corpo inteiro com FDG;
- Quinta-feira: 6 linfocintigrafia para detecção de ganglio sentinela, 12 cintigrafia miocárdica de perfusão em esforço/stress farmacológico, 1 tomografia cerebral com 123I-ioflupano, 2 cintigrafia miocárdica de perfusão em repouso, 10 PET - estudo corpo inteiro com FDG.

Tabela 6.2: Resultados provenientes dos três modelos em relação aos problemas de cada dia da semana.

Dia	BKS	MILP – δ			Modelo 1				Modelo 2				Modelo 3 <i>left-shifting</i>				Modelo 3 <i>enhanced left-shifting</i>				NEH	
		BRPD	BBRPD	TCT	BRPD	ARPD	TCT	ACT	BRPD	ARPD	TCT	ACT	BRPD	ARPD	TCT	ACT	BRPD	ARPD	TCT	ACT	BRPD	TCT
Segunda-Feira	487	0.00	-1.64	28800	2.87	4.76	104.3	18.1	2.26	4.02	94.3	12.5	2.46	3.23	11.2	1.5	1.64	2.57	20.1	3.0	8.21	0.17
Terça-Feira	441	0.00	0.00	73	0.00	12.15	84.9	14.1	0.45	0.98	130.8	26.0	0.00	0.77	9.9	1.2	0.00	0.79	13.7	1.8	4.54	0.16
Quarta-Feira	576	0.00	0.00	19547	11.28	138.57	50.0	8.1	4.17	4.83	105.8	13.4	3.12	3.59	6.2	0.8	0.00	1.00	8.9	1.4	10.24	0.06
Quinta-Feira	584	0.00	-4.56	28800	3.08	4.25	127.3	20.0	2.57	4.20	138.3	18.0	2.57	3.84	23.2	3.8	2.23	4.37	33.1	5.3	10.79	0.32
Média		0.00	-1.55	19305	4.31	39.93	91.6	15.1	2.36	3.51	117.3	17.5	2.04	2.86	12.6	1.8	0.97	2.18	19.0	2.9	8.44	0.18

Para cada modelo apresentado e a heurística NEH, observamos na tabela 6.2 a melhor solução encontrada (*BKS*) proveniente do modelo *MILP* – δ , o desvio relativo percentual da melhor solução (*BRPD*), o desvio relativo percentual do melhor *bound* encontrado (*BBRPD*), o desvio relativo percentual médio (*ARPD*), o tempo total de computação (*TCT*), e o tempo médio de computação (*ACT*). Os valores do desvio percentual foram obtidos através da fórmula:

$$RPD_h = \frac{C_{\max}^h - C_{\max}^B}{C_{\max}^B}$$

Tal que C_{\max}^h e C_{\max}^B são os valores de *makespan* obtidos pelo modelo *h* e *BKS*. Para a obtenção das soluções, cada modelo foi executado 100 vezes, para *BRPD* foi reportado o melhor valor encontrado e para *ARPD* utilizou-se a média do mínimo de 10 execuções. Por sua vez, *TCT* foi o tempo necessário para executar o modelo 100 vezes e para *ACT* utilizou-se a média do máximo de 10 execuções.

Apesar do modelo *MILP* – δ apresentar as melhores soluções o tempo computacional correspondente dificulta a sua utilização na prática. Também será difícil justificar a utilização do modelo 1 ou 2, sendo que estes em média apresentam soluções piores com maior tempo computacional. Por sua vez, não podemos dizer que modelo 3 com *left-shifting* ou com *enhanced left-shifting* apresentam soluções ou tempos estatisticamente diferentes, de acordo com a tabela 6.3, não rejeitando a hipótese nula, de que as duas populações pertencem à mesma distribuição, quer na solução ou no tempo computacional. A heurística NEH, considerando o baixo tempo computacional utilizado, torna-se numa escolha atraente se a obtenção de uma solução razoável e rápida seja necessário.

Tabela 6.3: Teste de Mann–Whitney para *BRPD*, *ARPD*, *TCT*, e *ACT* para o modelo 3 com *left shifting* e *enhanced left shifting*.

	Estatística	p-Value
<i>BRPD</i>	13	0.1832
<i>ARPD</i>	9	0.8857
<i>TCT</i>	5	0.4857
<i>ACT</i>	4	0.3429

Os valores referentes à melhor solução e ao tempo computacional total de cada modelo

com as várias combinações de variáveis estão presentes no apêndice (.2), bem como o modelo 3 utilizando *enchanced left shifting timetabling*.

6.1.5 Comparação com instâncias de *benchmark*

Nesta subsecção serão utilizados os modelos descritos anteriormente, utilizando os níveis identificados, na resolução de instâncias *benchmark*. Para tal foram utilizados 21 instâncias de pequenas dimensões, com entre 6 e 10 trabalhos, e com entre 6 e 10 operações. Estas instâncias foram escolhidas devido à comparação já existente em Sundar et al. **sundarHybridArtificialBee2017**, onde também encontramos o valor da melhor solução de cada instância.

Tabela 6.4: Resultados provenientes dos três modelos em relação às instâncias apresentadas em Sundar et al. **sundarHybridArtificialBee2017**.

Instância	BKS	Modelo 1				Modelo 2				Modelo 3				NEH	
		BRPD	ARPD	TCT	ACT	BRPD	ARPD	TCT	ACT	BRPD	ARPD	TCT	ACT	BRPD	TCT
Ft06	73	0.00	0.00	27.7	3.1	0.00	0.00	6.9	0.9	0.00	0.00	0.4	0.0	0.00	0.00
La01	971	0.00	9.58	62.5	9.7	0.00	5.26	58.8	7.6	0.41	4.00	6.4	0.8	31.10	0.92
La02	937	3.31	13.24	74.6	14.4	2.13	7.72	46.4	6.9	3.09	4.51	4.1	0.8	21.02	0.02
La03	820	6.10	14.28	59.5	11.5	2.56	7.29	40.5	6.0	5.12	6.05	3.3	0.4	23.90	0.01
La04	887	4.74	22.00	53.8	10.2	0.22	4.95	41.4	6.0	0.11	1.51	3.7	0.5	5.98	0.01
La05	777	0.90	12.59	51.2	7.9	3.73	8.11	34.5	5.0	0.90	2.32	2.9	0.4	10.81	0.01
Ft10	1607	12.44	29.56	150.4	25.3	1.62	4.84	256.8	41.9	0.00	4.92	21.0	3.0	4.73	0.06
Orb01	1615	4.27	20.18	193.5	39.4	0.00	3.22	239.8	35.7	0.00	1.94	23.8	3.2	11.95	0.06
Orb02	1485	6.13	30.42	117.6	26.0	2.22	4.09	205.2	29.2	2.22	2.22	20.2	3.0	2.66	0.06
Orb03	1599	2.31	7.81	206.1	35.3	0.25	3.00	273.4	39.5	0.00	1.36	22.5	3.3	17.01	0.07
Orb04	1653	12.16	28.68	135.9	28.0	4.42	6.64	289.2	44.6	4.46	5.89	26.7	4.3	21.34	0.09
Orb05	1365	8.06	15.52	110.0	19.9	6.15	8.81	236.5	37.3	0.44	3.31	16.2	2.5	18.02	0.05
Orb06	1555	0.00	10.15	153.4	24.1	0.00	2.24	294.7	42.6	0.00	0.06	25.2	3.5	22.96	0.08
Orb08	1319	2.35	29.00	146.8	27.4	0.00	2.52	186.4	30.0	0.00	0.00	16.6	2.2	7.66	0.04
Orb09	1445	6.02	33.88	119.8	29.5	0.62	5.97	225.2	33.6	4.08	7.31	20.7	3.0	24.57	0.07
Orb10	1557	1.73	20.48	111.2	18.5	1.48	4.01	292.4	46.2	0.00	2.58	25.3	3.7	13.04	0.11
La16	1575	4.51	13.74	134.0	25.2	0.00	3.59	283.8	44.2	1.17	4.72	22.2	3.1	6.73	0.06
La17	1371	10.72	31.82	94.9	14.5	1.97	5.92	170.0	23.5	4.30	6.16	18.4	2.9	17.36	0.07
La18	1417	9.81	25.79	122.4	27.0	6.35	9.56	240.8	35.7	6.35	6.63	20.6	3.0	19.97	0.07
La19	1482	1.48	20.29	112.7	19.2	1.28	6.13	228.1	34.2	4.45	6.39	23.2	3.2	28.41	0.07
La20	1526	10.62	51.03	132.1	24.1	5.37	8.29	284.5	43.5	0.00	1.80	24.0	3.5	10.94	0.08
Média		5.13	20.95	112.9	21.0	1.92	5.34	187.4	28.3	1.77	3.50	16.5	2.4	15.25	0.10

Apesar destes modelos não terem sido formulados para resolver as instâncias aqui apresentadas, sendo estes problemas apenas *No-Wait Job-Shop*, será possível observar que de qualquer forma apresentam boas soluções com tempo computacional razoável. Comparando os três modelos, será difícil justificar a utilização de qualquer um deles além do modelo 3, isto porque apresenta em médias as melhores soluções, com tempo computacional muito inferior aos outros dois.

Mesmo assim, os modelos aqui descritos são inferiores aos apresentados em Sundar et al. **sundarHybridArtificialBee2017** e em Ying & Lin [28]. O mesmo se verifica em relação ao tempo computacional, contudo esta comparação será menos relevante devido às diferentes linguagens de programação utilizadas e diferentes especificações de computadores.

BIBLIOGRAFIA

- [1] M. Meskarpour Amiri, M. Kazemian, Z. Motaghed e Z. Abdi, «Systematic Review of Factors Determining Health Care Expenditures,» *Health Policy and Technology*, vol. 10, n.º 2, p. 100-109, 2021-06, ISSN: 2211-8837. DOI: [10.1016/j.hlpt.2021.01.004](https://doi.org/10.1016/j.hlpt.2021.01.004).
acedido em 2025-05-16 (ver p. 1).
- [2] E. R. de Saúde, *ERS - Tempos Máximos de Resposta Garantidos (TMRG)*, <https://www.ers.pt/pt/utentes/p-frequentes/faq/tempos-maximos-de-resposta-garantidos-tmrg/>.
acedido em 2025-05-16 (ver p. 1).
- [3] A. Almén, «Trends in Diagnostic Nuclear Medicine in Sweden (2008–2023): Utilisation, Radiation Dose, and Methodological Insights,» *EJNMMI Physics*, vol. 12, n.º 1, p. 32, 2025-04, ISSN: 2197-7364. DOI: [10.1186/s40658-025-00747-2](https://doi.org/10.1186/s40658-025-00747-2).
acedido em 2025-05-21 (ver p. 1).
- [4] S. Dauzère-Pérès, J. Ding, L. Shen e K. Tamssaouet, «The Flexible Job Shop Scheduling Problem: A Review,» *European Journal of Operational Research*, vol. 314, n.º 2, pp. 409–432, 2024-04, Fala sobre as restrições extras que pode haver, e tem bastantes artigos citados com no-wait e blocking-, ISSN: 0377-2217. DOI: [10.1016/j.ejor.2023.05.017](https://doi.org/10.1016/j.ejor.2023.05.017).
acedido em 2025-04-06 (ver pp. 3, 8).
- [5] P. Brucker e R. Schlie, «Job-shop scheduling with multi-purpose machines,» *Computing*, vol. 45, n.º 4, pp. 369–375, 1990-12, ISSN: 1436-5057. DOI: [10.1007/BF02238804](https://doi.org/10.1007/BF02238804).
acedido em 2025-05-24 (ver p. 3).
- [6] P. Brandimarte, «Routing and scheduling in a flexible job shop by tabu search,» *Annals of Operations Research*, vol. 41, n.º 3, pp. 157–183, 1993-09, ISSN: 1572-9338. DOI: [10.1007/BF02023073](https://doi.org/10.1007/BF02023073).
acedido em 2025-04-09 (ver p. 3).
- [7] P. Brucker e J. Neyer, «Tabu-search for the multi-mode job-shop problem,» *Operations Research-Spektrum*, vol. 20, n.º 1, pp. 21–28, 1998-03, ISSN: 1436-6304. DOI: [10.1007/BF01545525](https://doi.org/10.1007/BF01545525).
acedido em 2025-05-26 (ver p. 3).

- [8] S. Dauzère-Pérès, W. Roux e J. B. Lasserre, «Multi-Resource Shop Scheduling with Resource Flexibility,» *European Journal of Operational Research*, vol. 107, n.º 2, pp. 289–305, 1998-06, ISSN: 0377-2217. DOI: [10.1016/S0377-2217\(97\)00341-X](https://doi.org/10.1016/S0377-2217(97)00341-X). acedido em 2025-04-08 (ver p. 3).
- [9] J. R. Callahan, *The Nothing Hot Delay Problem in the Production of Steel*. [Toronto], 1971, Open Library ID: OL19839735M (ver p. 4).
- [10] B. Jiang, Y. Ma, L. Chen, B. Huang, Y. Huang e L. Guan, «A Review on Intelligent Scheduling and Optimization for Flexible Job Shop,» *International Journal of Control, Automation and Systems*, vol. 21, n.º 10, pp. 3127–3150, 2023-10, ISSN: 2005-4092. DOI: [10.1007/s12555-023-0578-1](https://doi.org/10.1007/s12555-023-0578-1). acedido em 2025-04-10 (ver pp. 4, 7).
- [11] A. E. Ezugwu et al., «Metaheuristics: A comprehensive overview and classification along with bibliometric analysis,» *Artificial Intelligence Review*, vol. 54, n.º 6, pp. 4237–4316, 2021-08, ISSN: 1573-7462. DOI: [10.1007/s10462-020-09952-0](https://doi.org/10.1007/s10462-020-09952-0). acedido em 2025-07-10 (ver p. 5).
- [12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller e E. Teller, «Equation of State Calculations by Fast Computing Machines,» *The Journal of Chemical Physics*, vol. 21, n.º 6, pp. 1087–1092, 1953-06, ISSN: 0021-9606. DOI: [10.1063/1.1699114](https://doi.org/10.1063/1.1699114). acedido em 2025-05-28 (ver pp. 6, 19).
- [13] H. Xiong, S. Shi, D. Ren e J. Hu, «A Survey of Job Shop Scheduling Problem: The Types and Models,» *Computers & Operations Research*, vol. 142, p. 105731, 2022-06, ISSN: 0305-0548. DOI: [10.1016/j.cor.2022.105731](https://doi.org/10.1016/j.cor.2022.105731). acedido em 2025-04-16 (ver p. 7).
- [14] W.-Y. Ku e J. C. Beck, «Mixed Integer Programming Models for Job Shop Scheduling: A Computational Analysis,» *Computers & Operations Research*, vol. 73, pp. 165–173, 2016-09, ISSN: 0305-0548. DOI: [10.1016/j.cor.2016.04.006](https://doi.org/10.1016/j.cor.2016.04.006). acedido em 2025-07-07 (ver p. 7).
- [15] P. Fattahi, M. Saidi Mehrabad e F. Jolai, «Mathematical modeling and heuristic approaches to flexible job shop scheduling problems,» *Journal of Intelligent Manufacturing*, vol. 18, n.º 3, pp. 331–342, 2007-06, ISSN: 1572-8145. DOI: [10.1007/s10845-007-0026-8](https://doi.org/10.1007/s10845-007-0026-8). acedido em 2025-07-15 (ver p. 8).
- [16] C. Özgüven, L. Özbakır e Y. Yavuz, «Mathematical Models for Job-Shop Scheduling Problems with Routing and Process Plan Flexibility,» *Applied Mathematical Modelling*, vol. 34, n.º 6, pp. 1539–1548, 2010-06, ISSN: 0307-904X. DOI: [10.1016/j.apm.2009.09.002](https://doi.org/10.1016/j.apm.2009.09.002). acedido em 2025-05-26 (ver p. 8).
- [17] (PDF) *A competitive iterative procedure using a time-indexed model for solving flexible job shop scheduling problems*, https://www.researchgate.net/publication/311966224_A_competitive_iterative_indexed_model_for_solving_flexible_job_shop_scheduling_problems. acedido em 2025-07-14 (ver p. 8).

- [18] H. Samarghandi, «Solving the No-Wait Job Shop Scheduling Problem with Due Date Constraints: A Problem Transformation Approach,» *Computers & Industrial Engineering*, vol. 136, pp. 635–662, 2019-10, ISSN: 0360-8352. DOI: [10.1016/j.cie.2019.07.054](https://doi.org/10.1016/j.cie.2019.07.054). acessado em 2025-07-15 (ver p. 8).
- [19] R. Behmanesh e I. Rahimi, «Improved ant colony optimization for multi-resource job shop scheduling: A special case of transportation,» *Economic Computation and Economic Cybernetics Studies and Research*, vol. 55, n.º 4, pp. 277–294, 2021, ISSN: 0424-267X. DOI: [10.24818/18423264/55.4.21.18](https://doi.org/10.24818/18423264/55.4.21.18) (ver p. 8).
- [20] M. Nawaz, E. E. Ensore e I. Ham, «A Heuristic Algorithm for the M-Machine, n-Job Flow-Shop Sequencing Problem,» *Omega*, vol. 11, n.º 1, pp. 91–95, 1983-01, ISSN: 0305-0483. DOI: [10.1016/0305-0483\(83\)90088-9](https://doi.org/10.1016/0305-0483(83)90088-9). acessado em 2025-08-14 (ver p. 8).
- [21] K.-C. Ying e S.-W. Lin, «Minimizing Total Completion Time in the No-Wait Jobshop Scheduling Problem Using a Backtracking Metaheuristic,» *Computers & Industrial Engineering*, vol. 169, p. 108 238, 2022-07, ISSN: 0360-8352. DOI: [10.1016/j.cie.2022.108238](https://doi.org/10.1016/j.cie.2022.108238). acessado em 2025-08-14 (ver pp. 8, 9).
- [22] M. Xu, S. Zhang e G. Deng, «No-Wait Job Shop Scheduling Using a Population-Based Iterated Greedy Algorithm,» *Algorithms*, vol. 14, n.º 5, p. 145, 2021-05, ISSN: 1999-4893. DOI: [10.3390/a14050145](https://doi.org/10.3390/a14050145). acessado em 2025-08-14 (ver p. 8).
- [23] A. Aschauer, F. Roetzer, A. Steinboeck e A. Kugi, «An Efficient Algorithm for Scheduling a Flexible Job Shop with Blocking and No-Wait Constraints*,» *IFAC-PapersOnLine*, 20th IFAC World Congress, vol. 50, n.º 1, pp. 12 490–12 495, 2017-07, ISSN: 2405-8963. DOI: [10.1016/j.ifacol.2017.08.2056](https://doi.org/10.1016/j.ifacol.2017.08.2056). acessado em 2025-04-11 (ver p. 8).
- [24] W. Xia e Z. Wu, «An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems,» *Computers & Industrial Engineering*, vol. 48, n.º 2, pp. 409–425, 2005-03, ISSN: 0360-8352. DOI: [10.1016/j.cie.2005.01.018](https://doi.org/10.1016/j.cie.2005.01.018). acessado em 2025-08-14 (ver p. 8).
- [25] S. Jonathan, C. E. Nugraheni e L. Abednego, «A Whale Optimization Algorithm Based Solver for No-wait Flexible Flow Shop Scheduling Problems,» em *2024 Ninth International Conference on Informatics and Computing (ICIC)*, 2024-10, pp. 1–6. DOI: [10.1109/ICIC64337.2024.10956970](https://doi.org/10.1109/ICIC64337.2024.10956970). acessado em 2025-08-14 (ver p. 9).
- [26] Q. Perrachon, A.-L. Olteanu, M. Sevaux, S. Fréchengues e J.-F. Kerviche, «Industrial Multi-Resource Flexible Job Shop Scheduling with Partially Necessary Resources,» *European Journal of Operational Research*, vol. 320, n.º 2, pp. 309–327, 2025-01, ISSN: 0377-2217. DOI: [10.1016/j.ejor.2024.07.023](https://doi.org/10.1016/j.ejor.2024.07.023). acessado em 2025-08-13 (ver p. 9).

- [27] A. Caumont, P. Lacomme e N. Tchernev, «A Memetic Algorithm for the Job-Shop with Time-Lags,» *Computers & Operations Research*, Part Special Issue: Includes Selected Papers Presented at the ECCO'04 European Conference on Combinatorial Optimization, vol. 35, n.º 7, pp. 2331–2356, 2008-07, ISSN: 0305-0548. DOI: [10.1016/j.cor.2006.11.007](https://doi.org/10.1016/j.cor.2006.11.007). acedido em 2025-08-14 (ver p. 9).
- [28] K.-C. Ying e S.-W. Lin, «Solving No-Wait Job-Shop Scheduling Problems Using a Multi-Start Simulated Annealing with Bi-Directional Shift Timetabling Algorithm,» *Computers & Industrial Engineering*, vol. 146, p. 106 615, 2020-08, ISSN: 0360-8352. DOI: [10.1016/j.cie.2020.106615](https://doi.org/10.1016/j.cie.2020.106615). acedido em 2025-08-14 (ver pp. 27, 39).

1 Appendix:Tabelas

Tabela .5: Effect Estimates; Var.:Makespan; R-sqr=0,85919; Adj:0,82251; 5 3-level factors, 1 Blocks, 243 Runs; MS Residual=17657,26; DV: Makespan. Modelo 1, problema 1

Factors	Effect	Std.Err	t(48)	p	-95% Cnf.Limt	+95% Cnf.Limt	Coeff.	Std.Err.Coeff	-95% Cnf.Limt	+95% Cnf.Limt
Mean/Interc.	636,032	8,52430	74,6140	0,000000	619,219	652,845	636,032	8,52430	619,219	652,845
(1)Lk (L)	13,658	20,88017	0,6541	0,513821	-27,526	54,842	6,829	10,44009	-13,763	27,421
Lk (Q)	-10,417	18,08276	-0,5761	0,565253	-46,083	25,250	-5,208	9,04138	-23,042	12,625
(2)CP (L)	-340,247	20,88017	-16,2952	0,000000	-381,431	-299,063	-170,123	10,44009	-190,715	-149,531
CP (Q)	-163,604	18,08276	-9,0475	0,000000	-199,270	-127,937	-81,802	9,04138	-99,635	-63,969
(3)Alpha (L)	249,080	20,88017	11,9290	0,000000	207,896	290,264	124,540	10,44009	103,948	145,132
Alpha (Q)	-89,869	18,08276	-4,9698	0,000001	-125,535	-54,202	-44,934	9,04138	-62,767	-27,101
(4)P (L)	-118,298	20,88017	-5,6655	0,000000	-159,482	-77,114	-59,149	10,44009	-79,741	-38,557
P (Q)	-59,417	18,08276	-3,2858	0,001209	-95,083	-23,750	-29,708	9,04138	-47,542	-11,875
(5)p0 (L)	125,273	20,88017	5,9996	0,000000	84,089	166,457	62,636	10,44009	42,044	83,228
p0 (Q)	-27,413	18,08276	-1,5160	0,131171	-63,079	8,253	-13,706	9,04138	-31,540	4,127
1L by 2L	-45,407	25,57289	-1,7756	0,077382	-95,847	5,032	-22,704	12,78644	-47,924	2,516
1L by 2Q	-25,496	22,14677	-1,1512	0,251064	-69,179	18,186	-12,748	11,07338	-34,589	9,093
1Q by 2L	7,307	22,14677	0,3300	0,741795	-36,375	50,990	3,654	11,07338	-18,187	25,495
1Q by 2Q	3,433	19,17966	0,1790	0,858119	-34,397	41,263	1,717	9,58983	-17,198	20,632
1L by 3L	4,441	25,57289	0,1737	0,862323	-45,999	54,881	2,220	12,78644	-23,000	27,440
1L by 3Q	27,604	22,14677	1,2464	0,214137	-16,079	71,286	13,802	11,07338	-8,039	35,643
1Q by 3L	-4,280	22,14677	-0,1932	0,846976	-47,962	39,403	-2,140	11,07338	-23,981	19,701
1Q by 3Q	-4,022	19,17966	-0,2097	0,834114	-41,852	33,808	-2,011	9,58983	-20,926	16,904
1L by 4L	-61,296	25,57289	-2,3969	0,017492	-111,736	-10,856	-30,648	12,78644	-55,868	-5,428
1L by 4Q	-29,091	22,14677	-1,3135	0,190567	-72,773	14,591	-14,545	11,07338	-36,386	7,296
1Q by 4L	7,759	22,14677	0,3504	0,726455	-35,923	51,441	3,880	11,07338	-17,961	25,721
1Q by 4Q	3,664	19,17966	0,1910	0,848704	-34,166	41,494	1,832	9,58983	-17,083	20,747
1L by 5L	35,800	25,57289	1,3999	0,163151	-14,640	86,240	17,900	12,78644	-7,320	43,120
1L by 5Q	-4,652	22,14677	-0,2100	0,833854	-48,334	39,030	-2,326	11,07338	-24,167	19,515
1Q by 5L	1,704	22,14677	0,0769	0,938761	-41,979	45,386	0,852	11,07338	-20,989	22,693
1Q by 5Q	-12,406	19,17966	-0,6468	0,518529	-50,235	25,424	-6,203	9,58983	-25,118	12,712
2L by 3L	-372,176	25,57289	-14,5535	0,000000	-422,616	-321,736	-186,088	12,78644	-211,308	-160,868
2L by 3Q	134,521	22,14677	6,0741	0,000000	90,839	178,204	67,261	11,07338	45,420	89,102
2Q by 3L	-185,366	22,14677	-8,3699	0,000000	-229,048	-141,684	-92,683	11,07338	-114,524	-70,842
2Q by 3Q	66,885	19,17966	3,4873	0,000605	29,055	104,715	33,442	9,58983	14,527	52,357
2L by 4L	186,502	25,57289	7,2930	0,000000	136,062	236,942	93,251	12,78644	68,031	118,471
2L by 4Q	91,432	22,14677	4,1285	0,000054	47,750	135,115	45,716	11,07338	23,875	67,557
2Q by 4L	93,573	22,14677	4,2251	0,000037	49,891	137,255	46,787	11,07338	24,945	68,628
2Q by 4Q	45,562	19,17966	2,3756	0,018507	7,733	83,392	22,781	9,58983	3,866	41,696
2L by 5L	-187,011	25,57289	-7,3129	0,000000	-237,451	-136,571	-93,506	12,78644	-118,725	-68,286
2L by 5Q	40,696	22,14677	1,8376	0,067671	-2,986	84,379	20,348	11,07338	-1,493	42,189
2Q by 5L	-92,741	22,14677	-4,1876	0,000043	-136,423	-49,059	-46,370	11,07338	-68,211	-24,529
2Q by 5Q	20,197	19,17966	1,0531	0,293640	-17,633	58,027	10,099	9,58983	-8,816	29,014
3L by 4L	-109,100	25,57289	-4,2662	0,000031	-159,540	-58,660	-54,550	12,78644	-79,770	-29,330
3L by 4Q	-51,641	22,14677	-2,3318	0,020751	-95,323	-7,959	-25,820	11,07338	-47,661	-3,979
3Q by 4L	11,998	22,14677	0,5418	0,588615	-31,684	55,680	5,999	11,07338	-15,842	27,840
3Q by 4Q	3,328	19,17966	0,1735	0,862437	-34,502	41,158	1,664	9,58983	-17,251	20,579
3L by 5L	119,317	25,57289	4,6657	0,000006	68,877	169,757	59,658	12,78644	34,438	84,878
3L by 5Q	-19,582	22,14677	-0,8842	0,377688	-63,265	24,100	-9,791	11,07338	-31,632	12,050
3Q by 5L	-21,955	22,14677	-0,9913	0,322775	-65,637	21,728	-10,977	11,07338	-32,818	10,864
3Q by 5Q	2,487	19,17966	0,1297	0,896944	-35,342	40,317	1,244	9,58983	-17,671	20,159
4L by 5L	-71,591	25,57289	-2,7995	0,005641	-122,031	-21,151	-35,795	12,78644	-61,015	-10,575
4L by 5Q	16,668	22,14677	0,7526	0,452614	-27,015	60,350	8,334	11,07338	-13,507	30,175
4Q by 5L	-35,960	22,14677	-1,6237	0,106076	-79,642	7,722	-17,980	11,07338	-39,821	3,861
4Q by 5Q	6,540	19,17966	0,3410	0,733476	-31,290	44,370	3,270	9,58983	-15,645	22,185

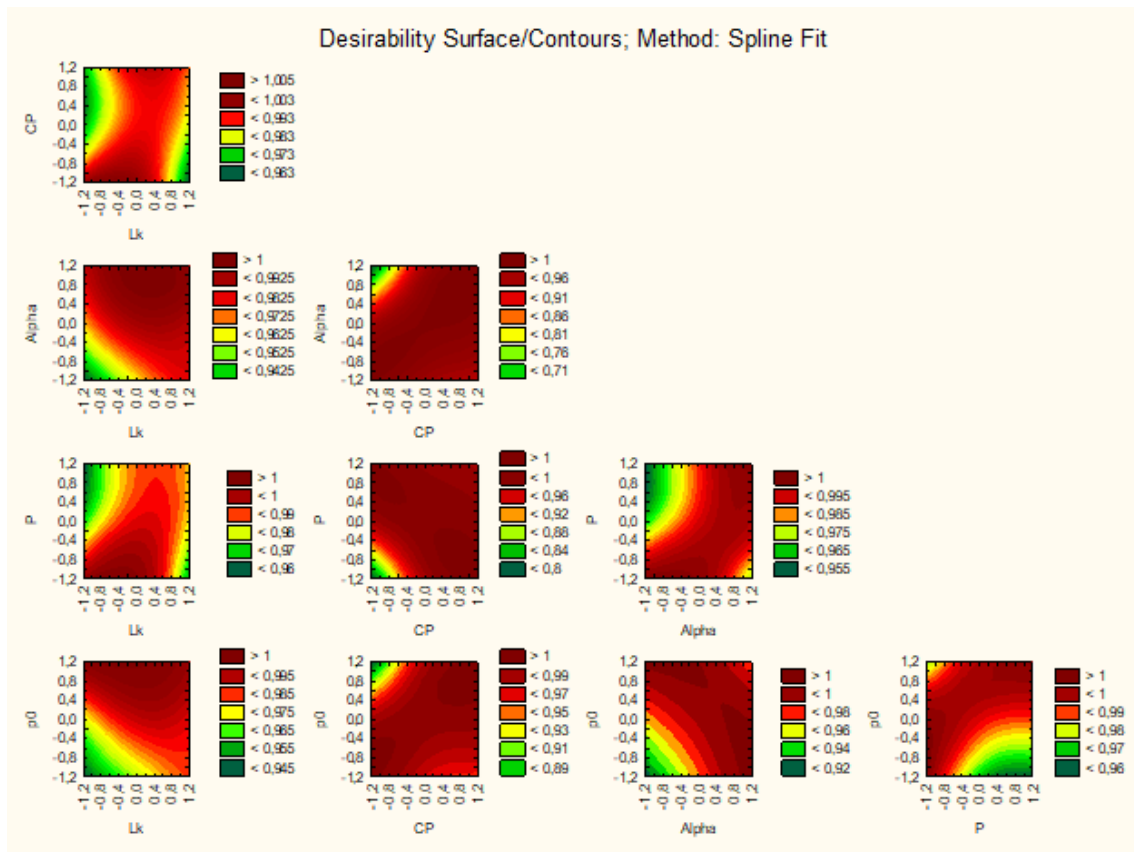


Figura 4: Gráfico de contorno das cinco variáveis, relativamente ao *makespan*, do modelo 1 do problema 1

Tabela .6: Effect Estimates; Var.:Tempo; R-sqr=0,98992; Adj:0,98729; 5 3-level factors, 1 Blocks, 243 Runs; MS Residual=1,459374; DV: Tempo. Modelo 1, problema 1

Factors	Effect	Std.Err	t(48)	p	-95% Cnf.Limt	+95% Cnf.Limt	Coeff.	Std.Err.Coeff	-95% Cnf.Limt	+95% Cnf.Limt
Mean/Interc.	12,13253	0,077496	156,5565	0,000000	11,97967	12,28538	12,13253	0,077496	11,97967	12,28538
(1)Lk (L)	13,62214	0,189826	71,7612	0,000000	13,24772	13,99655	6,81107	0,094913	6,62386	6,99827
Lk (Q)	1,16790	0,164394	7,1043	0,000000	0,84365	1,49215	0,58395	0,082197	0,42182	0,74607
(2)CP (L)	19,62236	0,189826	103,3702	0,000000	19,24795	19,99678	9,81118	0,094913	9,62398	9,99839
CP (Q)	1,66684	0,164394	10,1393	0,000000	1,34259	1,99109	0,83342	0,082197	0,67130	0,99555
(3)Alpha (L)	0,68372	0,189826	3,6018	0,000402	0,30931	1,05813	0,34186	0,094913	0,15465	0,52907
Alpha (Q)	-0,11882	0,164394	-0,7228	0,470684	-0,44307	0,20543	-0,05941	0,082197	-0,22154	0,10271
(4)P (L)	3,48910	0,189826	18,3805	0,000000	3,11468	3,86351	1,74455	0,094913	1,55734	1,93175
P (Q)	1,27850	0,164394	7,7771	0,000000	0,95425	1,60276	0,63925	0,082197	0,47713	0,80138
(5)p0 (L)	0,23432	0,189826	1,2344	0,218574	-0,14010	0,60873	0,11716	0,094913	-0,07005	0,30436
p0 (Q)	0,15427	0,164394	0,9384	0,349205	-0,16998	0,47852	0,07714	0,082197	-0,08499	0,23926
1L by 2L	10,28914	0,232488	44,2565	0,000000	9,83058	10,74770	5,14457	0,116244	4,91529	5,37385
1L by 2Q	1,77548	0,201341	8,8183	0,000000	1,37836	2,17261	0,88774	0,100670	0,68918	1,08630
1Q by 2L	1,16728	0,201341	5,7975	0,000000	0,77016	1,56440	0,58364	0,100670	0,38508	0,78220
1Q by 2Q	-0,23543	0,174366	-1,3502	0,178536	-0,57935	0,10849	-0,11772	0,087183	-0,28968	0,05424
1L by 3L	0,45376	0,232488	1,9517	0,052424	-0,00480	0,91232	0,22688	0,116244	-0,00240	0,45616
1L by 3Q	-0,09508	0,201341	-0,4722	0,637297	-0,49220	0,30204	-0,04754	0,100670	-0,24610	0,15102
1Q by 3L	-0,02559	0,201341	-0,1271	0,898978	-0,42272	0,37153	-0,01280	0,100670	-0,21136	0,18576
1Q by 3Q	0,17401	0,174366	0,9980	0,319545	-0,16991	0,51793	0,08701	0,087183	-0,08495	0,25897
1L by 4L	2,39385	0,232488	10,2966	0,000000	1,93529	2,85241	1,19692	0,116244	0,96764	1,42620
1L by 4Q	1,09805	0,201341	5,4537	0,000000	0,70092	1,49517	0,54902	0,100670	0,35046	0,74759
1Q by 4L	0,11748	0,201341	0,5835	0,560255	-0,27965	0,51460	0,05874	0,100670	-0,13982	0,25730
1Q by 4Q	-0,11948	0,174366	-0,6852	0,494026	-0,46340	0,22444	-0,05974	0,087183	-0,23170	0,11222
1L by 5L	0,17933	0,232488	0,7713	0,441453	-0,27923	0,63789	0,08966	0,116244	-0,13962	0,31894
1L by 5Q	0,22357	0,201341	1,1104	0,268218	-0,17356	0,62069	0,11178	0,100670	-0,08678	0,31035
1Q by 5L	-0,11632	0,201341	-0,5777	0,564118	-0,51345	0,28080	-0,05816	0,100670	-0,25672	0,14040
1Q by 5Q	0,02635	0,174366	0,1511	0,880036	-0,31757	0,37027	0,01318	0,087183	-0,15878	0,18514
2L by 3L	1,74960	0,232488	7,5255	0,000000	1,29104	2,20816	0,87480	0,116244	0,64552	1,10408
2L by 3Q	-0,54684	0,201341	-2,7160	0,007211	-0,94397	-0,14972	-0,27342	0,100670	-0,47198	-0,07486
2Q by 3L	0,75124	0,201341	3,7312	0,000251	0,35411	1,14836	0,37562	0,100670	0,17706	0,57418
2Q by 3Q	-0,33750	0,174366	-1,9356	0,054385	-0,68142	0,00642	-0,16875	0,087183	-0,34071	0,00321
2L by 4L	2,25720	0,232488	9,7089	0,000000	1,79864	2,71576	1,12860	0,116244	0,89932	1,35788
2L by 4Q	0,96528	0,201341	4,7942	0,000003	0,56815	1,36240	0,48264	0,100670	0,28408	0,68120
2Q by 4L	0,53284	0,201341	2,6465	0,008809	0,13572	0,92997	0,26642	0,100670	0,06786	0,46498
2Q by 4Q	-0,01009	0,174366	-0,0579	0,953909	-0,35401	0,33383	-0,00505	0,087183	-0,17701	0,16691
2L by 5L	0,40344	0,232488	1,7353	0,084290	-0,05512	0,86200	0,20172	0,116244	-0,02756	0,43100
2L by 5Q	0,16409	0,201341	0,8150	0,416084	-0,23303	0,56122	0,08205	0,100670	-0,11652	0,28061
2Q by 5L	0,07408	0,201341	0,3680	0,713316	-0,32304	0,47121	0,03704	0,100670	-0,16152	0,23560
2Q by 5Q	-0,25198	0,174366	-1,4451	0,150051	-0,59590	0,09194	-0,12599	0,087183	-0,29795	0,04597
3L by 4L	0,33507	0,232488	1,4412	0,151145	-0,12349	0,79363	0,16754	0,116244	-0,06174	0,39682
3L by 4Q	0,23495	0,201341	1,1669	0,244687	-0,16217	0,63207	0,11748	0,100670	-0,08109	0,31604
3Q by 4L	-0,21669	0,201341	-1,0763	0,283162	-0,61382	0,18043	-0,10835	0,100670	-0,30691	0,09021
3Q by 4Q	0,07438	0,174366	0,4266	0,670168	-0,26954	0,41830	0,03719	0,087183	-0,13477	0,20915
3L by 5L	0,50616	0,232488	2,1771	0,030691	0,04760	0,96472	0,25308	0,116244	0,02380	0,48236
3L by 5Q	-0,29040	0,201341	-1,4423	0,150845	-0,68752	0,10673	-0,14520	0,100670	-0,34376	0,05336
3Q by 5L	-0,34815	0,201341	-1,7292	0,085384	-0,74528	0,04897	-0,17408	0,100670	-0,37264	0,02448
3Q by 5Q	-0,19125	0,174366	-1,0968	0,274093	-0,53517	0,15267	-0,09562	0,087183	-0,26758	0,07634
4L by 5L	0,19665	0,232488	0,8458	0,398697	-0,26191	0,65521	0,09832	0,116244	-0,13096	0,32760
4L by 5Q	-0,28064	0,201341	-1,3938	0,164975	-0,67776	0,11649	-0,14032	0,100670	-0,33888	0,05824
4Q by 5L	0,16679	0,201341	0,8284	0,408467	-0,23033	0,56392	0,08340	0,100670	-0,11517	0,28196
4Q by 5Q	-0,02010	0,174366	-0,1153	0,908333	-0,36402	0,32382	-0,01005	0,087183	-0,18201	0,16191

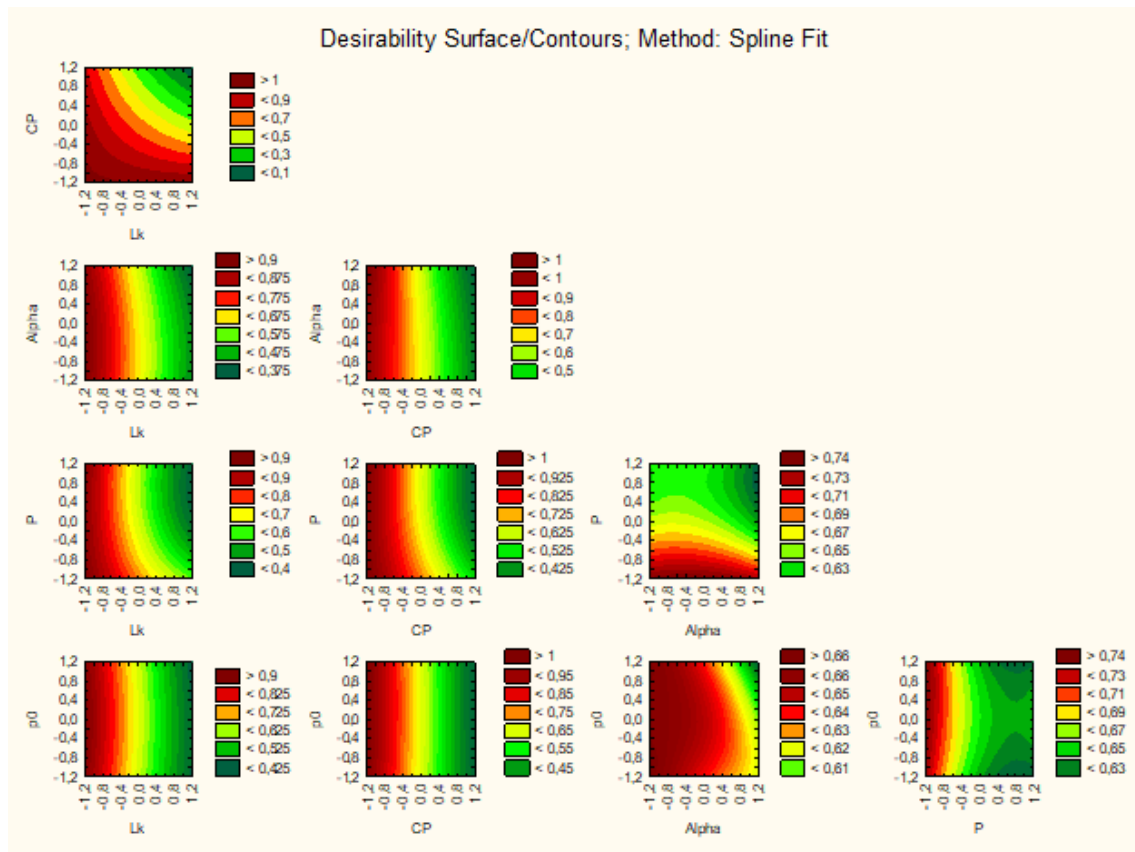


Figura .5: Gráfico de contorno das cinco variáveis, relativamente ao tempo, do modelo 1 do problema 1

Tabela .7: Effect Estimates; Var.:Makespan; R-sqr=0,79905; Adj:0,66508 ; 4 3-level factors, 1 Blocks, 81 Runs; MS Residual=29,81784; DV: Makespan. Modelo 2, problema 1

Factors	Effect	Std.Err	t(48)	p	-95% Cnf.Limt	+95% Cnf.Limt	Coeff.	Std.Err.Coeff	-95% Cnf.Limt	+95% Cnf.Limt
Mean/Interc.	512,7222	0,606730	845,0581	0,000000	511,5023	513,9421	512,7222	0,606730	511,5023	513,9421
(1)Lk (L)	-9,3259	1,486179	-6,2751	0,000000	-12,3141	-6,3378	-4,6630	0,743090	-6,1570	-3,1689
Lk (Q)	-2,1111	1,287069	-1,6402	0,107492	-4,6989	0,4767	-1,0556	0,643535	-2,3495	0,2384
(2)CP (L)	-10,9111	1,486179	-7,3417	0,000000	-13,8993	-7,9229	-5,4556	0,743090	-6,9496	-3,9615
CP (Q)	-4,9667	1,287069	-3,8589	0,000340	-7,5545	-2,3788	-2,4833	0,643535	-3,7772	-1,1894
(3)Alpha (L)	0,5667	1,486179	0,3813	0,704670	-2,4215	3,5548	0,2833	0,743090	-1,2107	1,7774
Alpha (Q)	-1,2167	1,287069	-0,9453	0,349240	-3,8045	1,3712	-0,6083	0,643535	-1,9022	0,6856
(4)p0 (L)	0,6815	1,486179	0,4585	0,648630	-2,3067	3,6696	0,3407	0,743090	-1,1533	1,8348
p0 (Q)	-2,0889	1,287069	-1,6230	0,111144	-4,6767	0,4989	-1,0444	0,643535	-2,3384	0,2495
1L by 2L	7,4889	1,820190	4,1143	0,000152	3,8292	11,1486	3,7444	0,910095	1,9146	5,5743
1L by 2Q	3,2444	1,576331	2,0582	0,045020	0,0750	6,4139	1,6222	0,788166	0,0375	3,2069
1Q by 2L	2,3500	1,576331	1,4908	0,142556	-0,8194	5,5194	1,1750	0,788166	-0,4097	2,7597
1Q by 2Q	0,9083	1,365143	0,6654	0,508995	-1,8365	3,6531	0,4542	0,682571	-0,9182	1,8266
1L by 3L	-2,2556	1,820190	-1,2392	0,221302	-5,9153	1,4042	-1,1278	0,910095	-2,9576	0,7021
1L by 3Q	0,5111	1,576331	0,3242	0,747164	-2,6583	3,6805	0,2556	0,788166	-1,3292	1,8403
1Q by 3L	-0,6667	1,576331	-0,4229	0,674240	-3,8361	2,5028	-0,3333	0,788166	-1,9180	1,2514
1Q by 3Q	-0,0417	1,365143	-0,0305	0,975777	-2,7865	2,7031	-0,0208	0,682571	-1,3932	1,3516
1L by 4L	-2,9222	1,820190	-1,6054	0,114955	-6,5820	0,7375	-1,4611	0,910095	-3,2910	0,3688
1L by 4Q	0,2444	1,576331	0,1551	0,877415	-2,9250	3,4139	0,1222	0,788166	-1,4625	1,7069
1Q by 4L	0,3889	1,576331	0,2467	0,806189	-2,7805	3,5583	0,1944	0,788166	-1,3903	1,7792
1Q by 4Q	0,2250	1,365143	0,1648	0,869779	-2,5198	2,9698	0,1125	0,682571	-1,2599	1,4849
2L by 3L	-5,8333	1,820190	-3,2048	0,002404	-9,4931	-2,1736	-2,9167	0,910095	-4,7465	-1,0868
2L by 3Q	3,3500	1,576331	2,1252	0,038745	0,1806	6,5194	1,6750	0,788166	0,0903	3,2597
2Q by 3L	-2,9667	1,576331	-1,8820	0,065905	-6,1361	0,2028	-1,4833	0,788166	-3,0680	0,1014
2Q by 3Q	1,6500	1,365143	1,2087	0,232712	-1,0948	4,3948	0,8250	0,682571	-0,5474	2,1974
2L by 4L	-5,1444	1,820190	-2,8263	0,006844	-8,8042	-1,4847	-2,5722	0,910095	-4,4021	-0,7424
2L by 4Q	2,9167	1,576331	1,8503	0,070432	-0,2528	6,0861	1,4583	0,788166	-0,1264	3,0430
2Q by 4L	-2,5111	1,576331	-1,5930	0,117723	-5,6805	0,6583	-1,2556	0,788166	-2,8403	0,3292
2Q by 4Q	1,4667	1,365143	1,0744	0,288029	-1,2781	4,2115	0,7333	0,682571	-0,6391	2,1057
3L by 4L	3,7111	1,820190	2,0389	0,046992	0,0514	7,3708	1,8556	0,910095	0,0257	3,6854
3L by 4Q	-4,2167	1,576331	-2,6750	0,010189	-7,3861	-1,0472	-2,1083	0,788166	-3,6930	-0,5236
3Q by 4L	-1,5944	1,576331	-1,0115	0,316853	-4,7639	1,5750	-0,7972	0,788166	-2,3819	0,7875
3Q by 4Q	1,0167	1,365143	0,7447	0,460064	-1,7281	3,7615	0,5083	0,682571	-0,8641	1,8807

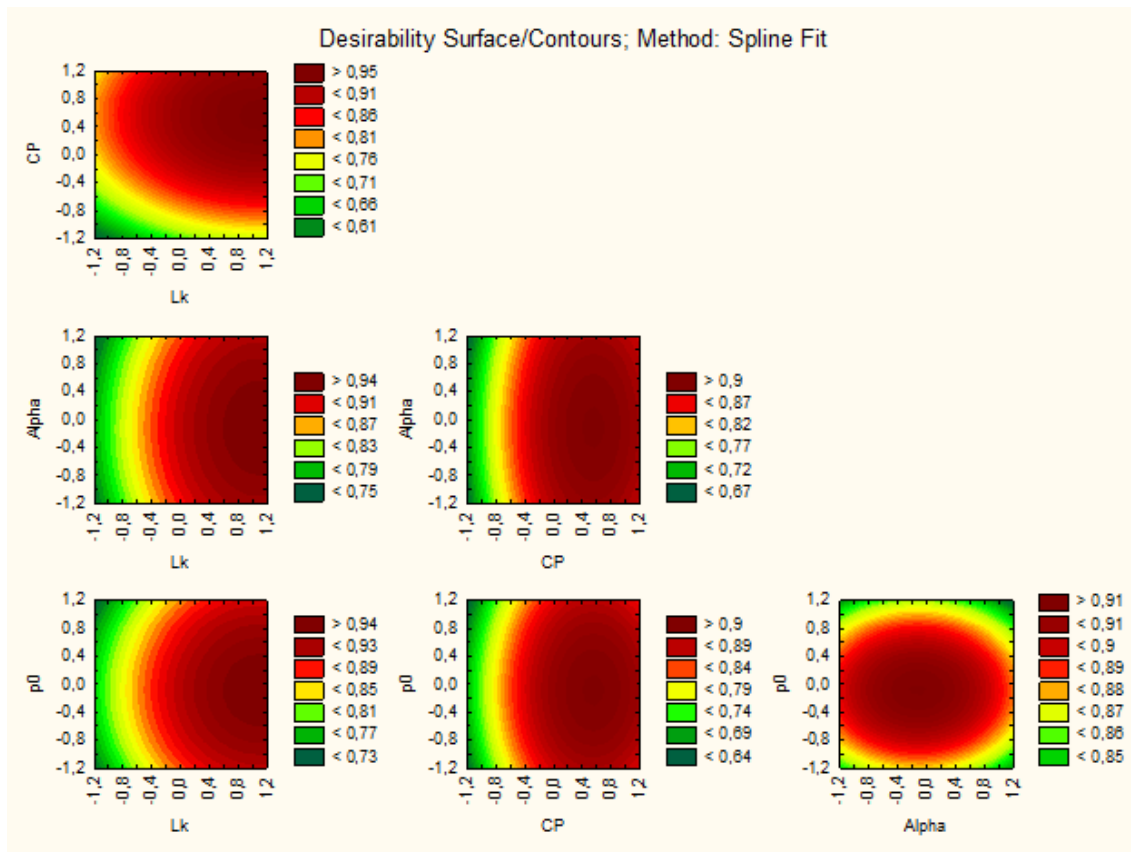


Figura .6: Gráfico de contorno das quatro variáveis relevantes, relativamente ao *makespan*, do modelo 2 do problema 1

Tabela .8: Effect Estimates; Var.:Tempo; R-sqr=0,97985; Adj:0,96642; 4 3-level factors, 1 Blocks, 81 Runs; MS Residual=5,912985; DV: Tempo. Modelo 2, problema 1

Factors	Effect	Std.Err	t(48)	p	-95% Cnf.Limt	+95% Cnf.Limt	Coeff.	Std.Err.Coeff	-95% Cnf.Limt	+95% Cnf.Limt
Mean/Interc.	19,94905	0,270185	73,83485	0,000000	19,40581	20,49229	19,94905	0,270185	19,40581	20,49229
(1)Lk (L)	21,72384	0,661815	32,82465	0,000000	20,39317	23,05451	10,86192	0,330907	10,19659	11,52725
Lk (Q)	0,72651	0,573148	1,26758	0,211063	-0,42588	1,87890	0,36326	0,286574	-0,21294	0,93945
(2)CP (L)	19,97500	0,661815	30,18215	0,000000	18,64433	21,30566	9,98750	0,330907	9,32216	10,65283
CP (Q)	2,84850	0,573148	4,96991	0,000009	1,69611	4,00089	1,42425	0,286574	0,84805	2,00045
(3)Alpha (L)	3,31104	0,661815	5,00297	0,000008	1,98037	4,64170	1,65552	0,330907	0,99018	2,32085
Alpha (Q)	-1,14631	0,573148	-2,00001	0,051174	-2,29870	0,00609	-0,57315	0,286574	-1,14935	0,00304
(4)p0 (L)	2,11030	0,661815	3,18866	0,002517	0,77964	3,44097	1,05515	0,330907	0,38982	1,72049
p0 (Q)	-1,01754	0,573148	-1,77534	0,082182	-2,16993	0,13486	-0,50877	0,286574	-1,08496	0,06743
1L by 2L	11,26922	0,810554	13,90310	0,000000	9,63949	12,89895	5,63461	0,405277	4,81975	6,44947
1L by 2Q	1,29787	0,701961	1,84892	0,070633	-0,11352	2,70926	0,64894	0,350980	-0,05676	1,35463
1Q by 2L	0,03951	0,701961	0,05629	0,955346	-1,37187	1,45090	0,01976	0,350980	-0,68594	0,72545
1Q by 2Q	-0,11595	0,607916	-0,19073	0,849543	-1,33824	1,10635	-0,05797	0,303958	-0,66912	0,55318
1L by 3L	4,36745	0,810554	5,38822	0,000002	2,73772	5,99718	2,18372	0,405277	1,36886	2,99859
1L by 3Q	-1,49005	0,701961	-2,12270	0,038963	-2,90144	-0,07867	-0,74503	0,350980	-1,45072	-0,03933
1Q by 3L	-0,62145	0,701961	-0,88530	0,380406	-2,03284	0,78994	-0,31072	0,350980	-1,01642	0,39497
1Q by 3Q	-0,04126	0,607916	-0,06787	0,946169	-1,26356	1,18104	-0,02063	0,303958	-0,63178	0,59052
1L by 4L	2,06900	0,810554	2,55257	0,013929	0,43927	3,69873	1,03450	0,405277	0,21963	1,84936
1L by 4Q	-0,88475	0,701961	-1,26039	0,213621	-2,29613	0,52664	-0,44237	0,350980	-1,14807	0,26332
1Q by 4L	-0,12064	0,701961	-0,17186	0,864271	-1,53202	1,29075	-0,06032	0,350980	-0,76601	0,64537
1Q by 4Q	-0,25861	0,607916	-0,42540	0,672449	-1,48090	0,96369	-0,12930	0,303958	-0,74045	0,48185
2L by 3L	2,11549	0,810554	2,60993	0,012043	0,48576	3,74522	1,05774	0,405277	0,24288	1,87261
2L by 3Q	-0,84142	0,701961	-1,19868	0,236538	-2,25281	0,56996	-0,42071	0,350980	-1,12641	0,28498
2Q by 3L	0,88890	0,701961	1,26630	0,211515	-0,52249	2,30028	0,44445	0,350980	-0,26125	1,15014
2Q by 3Q	-0,55794	0,607916	-0,91779	0,363316	-1,78024	0,66436	-0,27897	0,303958	-0,89012	0,33218
2L by 4L	0,63012	0,810554	0,77740	0,440738	-0,99960	2,25985	0,31506	0,405277	-0,49980	1,12993
2L by 4Q	-0,42321	0,701961	-0,60290	0,549415	-1,83460	0,98818	-0,21161	0,350980	-0,91730	0,49409
2Q by 4L	0,48816	0,701961	0,69542	0,490147	-0,92323	1,89954	0,24408	0,350980	-0,46162	0,94977
2Q by 4Q	-0,17075	0,607916	-0,28088	0,780006	-1,39305	1,05154	-0,08538	0,303958	-0,69653	0,52577
3L by 4L	3,86791	0,810554	4,77193	0,000017	2,23818	5,49763	1,93395	0,405277	1,11909	2,74882
3L by 4Q	-0,61189	0,701961	-0,87169	0,387719	-2,02328	0,79950	-0,30594	0,350980	-1,01164	0,39975
3Q by 4L	-0,88585	0,701961	-1,26197	0,213058	-2,29724	0,52553	-0,44293	0,350980	-1,14862	0,26277
3Q by 4Q	0,41730	0,607916	0,68645	0,495734	-0,80499	1,63960	0,20865	0,303958	-0,40250	0,81980

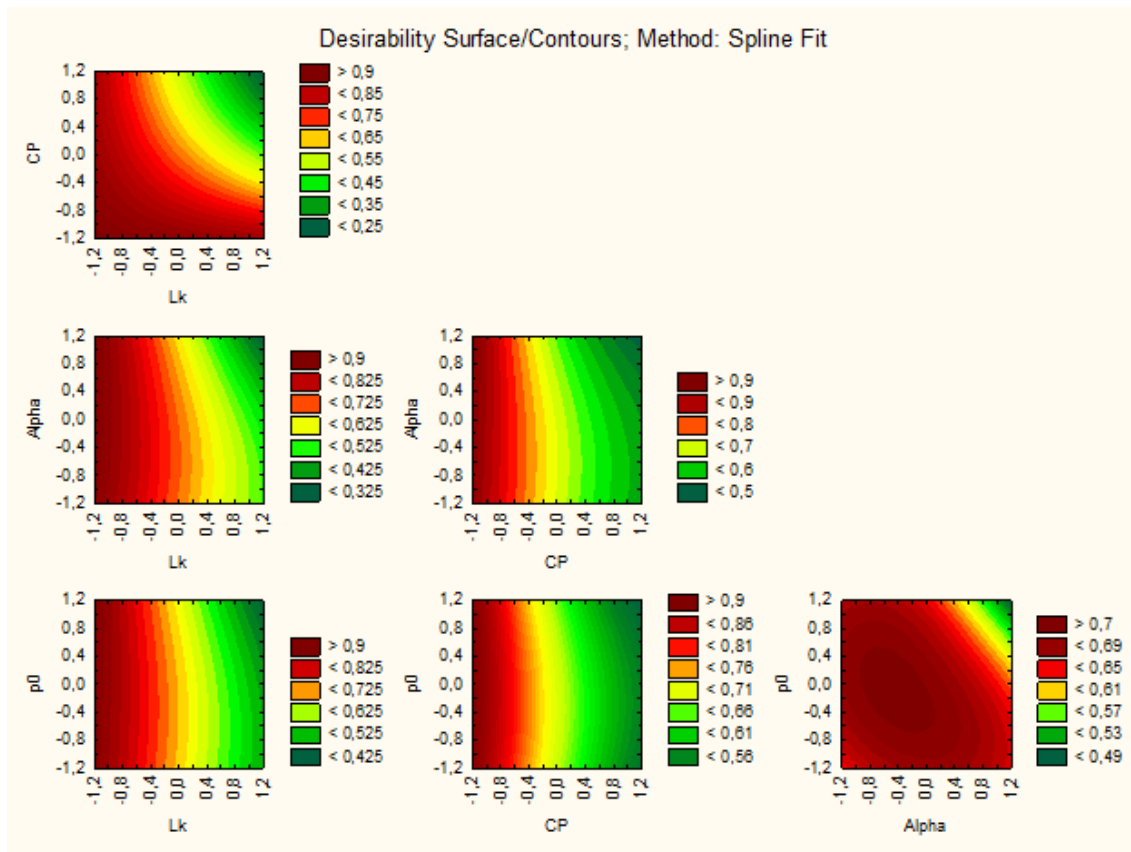


Figura .7: Gráfico de contorno das quatro variáveis relevantes, relativamente ao tempo, do modelo 2 do problema 1

Tabela .9: Effect Estimates; Var.:Makespan; R-sqr=0,97227; Adj:0,95378; 4 3-level factors, 1 Blocks, 81 Runs; MS Residual=1,415756; DV: Makespan. Modelo 3, problema 1

Factors	Effect	Std.Err	t(48)	p	-95% Cnf.Limt	+95% Cnf.Limt	Coeff.	Std.Err.Coeff	-95% Cnf.Limt	+95% Cnf.Limt
Mean/Interc.	501,4914	0,132206	3793,252	0,000000	501,2255	501,7572	501,4914	0,132206	501,2255	501,7572
(1)Lk (L)	-2,0185	0,323838	-6,233	0,000000	-2,6696	-1,3674	-1,0093	0,161919	-1,3348	-0,6837
Lk (Q)	-0,3537	0,280452	-1,261	0,213335	-0,9176	0,2102	-0,1769	0,140226	-0,4588	0,1051
(2)CP (L)	-9,6630	0,323838	-29,839	0,000000	-10,3141	-9,0118	-4,8315	0,161919	-5,1570	-4,5059
CP (Q)	-3,7093	0,280452	-13,226	0,000000	-4,2731	-3,1454	-1,8546	0,140226	-2,1366	-1,5727
(3)Alpha (L)	4,6111	0,323838	14,239	0,000000	3,9600	5,2622	2,3056	0,161919	1,9800	2,6311
Alpha (Q)	-0,0759	0,280452	-0,271	0,787762	-0,6398	0,4880	-0,0380	0,140226	-0,3199	0,2440
(4)p0 (L)	2,8481	0,323838	8,795	0,000000	2,1970	3,4993	1,4241	0,161919	1,0985	1,7496
p0 (Q)	-0,0870	0,280452	-0,310	0,757641	-0,6509	0,4768	-0,0435	0,140226	-0,3255	0,2384
1L by 2L	1,6611	0,396619	4,188	0,000120	0,8637	2,4586	0,8306	0,198309	0,4318	1,2293
1L by 2Q	0,5306	0,343482	1,545	0,129001	-0,1601	1,2212	0,2653	0,171741	-0,0800	0,6106
1Q by 2L	-0,0861	0,343482	-0,251	0,803116	-0,7767	0,6045	-0,0431	0,171741	-0,3884	0,3023
1Q by 2Q	-0,1431	0,297464	-0,481	0,632760	-0,7411	0,4550	-0,0715	0,148732	-0,3706	0,2275
1L by 3L	-0,3889	0,396619	-0,981	0,331750	-1,1863	0,4086	-0,1944	0,198309	-0,5932	0,2043
1L by 3Q	0,0556	0,343482	0,162	0,872188	-0,6351	0,7462	0,0278	0,171741	-0,3175	0,3731
1Q by 3L	-0,1167	0,343482	-0,340	0,735595	-0,8073	0,5739	-0,0583	0,171741	-0,4036	0,2870
1Q by 3Q	0,1944	0,297464	0,654	0,516441	-0,4036	0,7925	0,0972	0,148732	-0,2018	0,3963
1L by 4L	0,0833	0,396619	0,210	0,834472	-0,7141	0,8808	0,0417	0,198309	-0,3571	0,4404
1L by 4Q	-0,3528	0,343482	-1,027	0,309538	-1,0434	0,3378	-0,1764	0,171741	-0,5217	0,1689
1Q by 4L	0,3639	0,343482	1,059	0,294714	-0,3267	1,0545	0,1819	0,171741	-0,1634	0,5273
1Q by 4Q	0,0403	0,297464	0,135	0,892859	-0,5578	0,6384	0,0201	0,148732	-0,2789	0,3192
2L by 3L	-5,0389	0,396619	-12,705	0,000000	-5,8363	-4,2414	-2,5194	0,198309	-2,9182	-2,1207
2L by 3Q	-0,8528	0,343482	-2,483	0,016586	-1,5434	-0,1622	-0,4264	0,171741	-0,7717	-0,0811
2Q by 3L	-1,0750	0,343482	-3,130	0,002975	-1,7656	-0,3844	-0,5375	0,171741	-0,8828	-0,1922
2Q by 3Q	-1,1764	0,297464	-3,955	0,000252	-1,7745	-0,5783	-0,5882	0,148732	-0,8872	-0,2891
2L by 4L	-3,0111	0,396619	-7,592	0,000000	-3,8086	-2,2137	-1,5056	0,198309	-1,9043	-1,1068
2L by 4Q	0,1389	0,343482	0,404	0,687747	-0,5517	0,8295	0,0694	0,171741	-0,2759	0,4148
2Q by 4L	-0,6278	0,343482	-1,828	0,073816	-1,3184	0,0628	-0,3139	0,171741	-0,6592	0,0314
2Q by 4Q	-0,0806	0,297464	-0,271	0,787700	-0,6786	0,5175	-0,0403	0,148732	-0,3393	0,2588
3L by 4L	1,7000	0,396619	4,286	0,000087	0,9025	2,4975	0,8500	0,198309	0,4513	1,2487
3L by 4Q	0,0333	0,343482	0,097	0,923094	-0,6573	0,7239	0,0167	0,171741	-0,3286	0,3620
3Q by 4L	0,1889	0,343482	0,550	0,584921	-0,5017	0,8795	0,0944	0,171741	-0,2509	0,4398
3Q by 4Q	0,0694	0,297464	0,233	0,816401	-0,5286	0,6675	0,0347	0,148732	-0,2643	0,3338

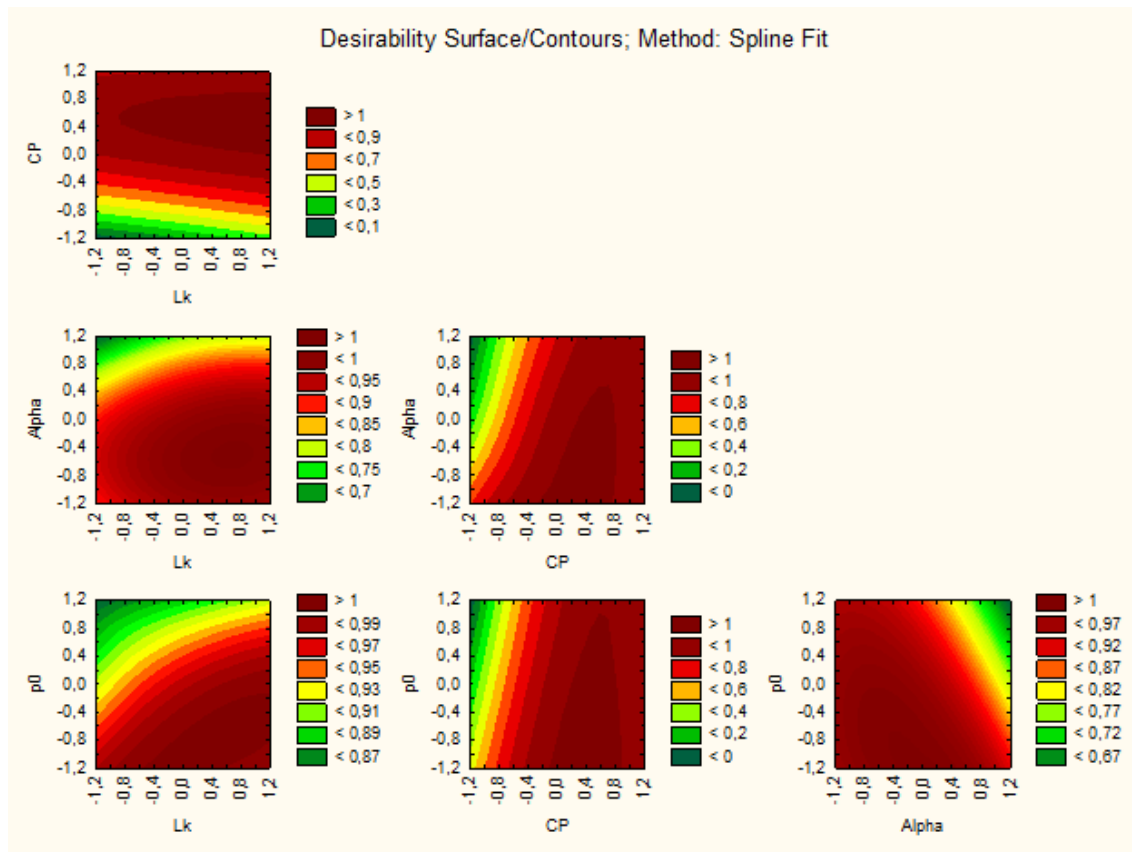


Figura .8: Gráfico de contorno das quatro variáveis relevantes, relativamente ao *makespan*, do modelo 3 do problema 1

Tabela .10: Effect Estimates; Var.:Tempo; R-sqr=0,97945; Adj:0,96576; 4 3-level factors, 1 Blocks, 81 Runs; MS Residual=10,33885; DV: Tempo. Modelo 3, problema 1

Factors	Effect	Std.Err	t(48)	p	-95% Cnf.Limt	+95% Cnf.Limt	Coeff.	Std.Err.Coeff	-95% Cnf.Limt	+95% Cnf.Limt
Mean/Interc.	18,67782	0,357268	52,27963	0,000000	17,95948	19,39615	18,67782	0,357268	17,95948	19,39615
(1)Lk (L)	23,68873	0,875123	27,06902	0,000000	21,92917	25,44828	11,84436	0,437562	10,96459	12,72414
Lk (Q)	0,10397	0,757879	0,13718	0,891460	-1,41985	1,62779	0,05198	0,378939	-0,70993	0,81389
(2)CP (L)	28,05016	0,875123	32,05281	0,000000	26,29061	29,80971	14,02508	0,437562	13,14530	14,90486
CP (Q)	2,21749	0,757879	2,92592	0,005233	0,69368	3,74131	1,10875	0,378939	0,34684	1,87066
(3)Alpha (L)	8,28376	0,875123	9,46582	0,000000	6,52421	10,04331	4,14188	0,437562	3,26210	5,02166
Alpha (Q)	-3,11745	0,757879	-4,11338	0,000152	-4,64127	-1,59363	-1,55872	0,378939	-2,32063	-0,79682
(4)p0 (L)	2,35684	0,875123	2,69316	0,009720	0,59729	4,11640	1,17842	0,437562	0,29865	2,05820
p0 (Q)	-0,32421	0,757879	-0,42778	0,670724	-1,84802	1,19961	-0,16210	0,378939	-0,92401	0,59981
1L by 2L	17,15184	1,071803	16,00280	0,000000	14,99684	19,30685	8,57592	0,535901	7,49842	9,65342
1L by 2Q	1,94505	0,928208	2,09549	0,041427	0,07876	3,81134	0,97252	0,464104	0,03938	1,90567
1Q by 2L	0,34062	0,928208	0,36697	0,715257	-1,52567	2,20691	0,17031	0,464104	-0,76283	1,10345
1Q by 2Q	-0,27891	0,803852	-0,34697	0,730133	-1,89516	1,33734	-0,13945	0,401926	-0,94758	0,66867
1L by 3L	6,60764	1,071803	6,16497	0,000000	4,45263	8,76264	3,30382	0,535901	2,22632	4,38132
1L by 3Q	-2,15015	0,928208	-2,31645	0,024848	-4,01644	-0,28386	-1,07508	0,464104	-2,00822	-0,14193
1Q by 3L	-0,57409	0,928208	-0,61850	0,539172	-2,44038	1,29219	-0,28705	0,464104	-1,22019	0,64610
1Q by 3Q	-0,38687	0,803852	-0,48127	0,632512	-2,00312	1,22938	-0,19343	0,401926	-1,00156	0,61469
1L by 4L	1,69692	1,071803	1,58324	0,119934	-0,45808	3,85193	0,84846	0,535901	-0,22904	1,92596
1L by 4Q	-0,30746	0,928208	-0,33124	0,741905	-2,17375	1,55883	-0,15373	0,464104	-1,08687	0,77941
1Q by 4L	0,10229	0,928208	0,11020	0,912713	-1,76400	1,96857	0,05114	0,464104	-0,88200	0,98429
1Q by 4Q	-0,01460	0,803852	-0,01816	0,985586	-1,63085	1,60165	-0,00730	0,401926	-0,81543	0,80083
2L by 3L	9,26165	1,071803	8,64119	0,000000	7,10665	11,41665	4,63082	0,535901	3,55332	5,70833
2L by 3Q	-3,39010	0,928208	-3,65231	0,000642	-5,25639	-1,52381	-1,69505	0,464104	-2,62819	-0,76191
2Q by 3L	0,71289	0,928208	0,76803	0,446234	-1,15340	2,57918	0,35644	0,464104	-0,57670	1,28959
2Q by 3Q	0,30248	0,803852	0,37629	0,708356	-1,31377	1,91874	0,15124	0,401926	-0,65688	0,95937
2L by 4L	2,67906	1,071803	2,49958	0,015906	0,52406	4,83407	1,33953	0,535901	0,26203	2,41703
2L by 4Q	-0,68407	0,928208	-0,73698	0,464724	-2,55036	1,18222	-0,34203	0,464104	-1,27518	0,59111
2Q by 4L	-0,22473	0,928208	-0,24211	0,809723	-2,09102	1,64155	-0,11237	0,464104	-1,04551	0,82078
2Q by 4Q	0,39673	0,803852	0,49354	0,623881	-1,21952	2,01299	0,19837	0,401926	-0,60976	1,00649
3L by 4L	1,37885	1,071803	1,28648	0,204446	-0,77615	3,53386	0,68943	0,535901	-0,38807	1,76693
3L by 4Q	0,29705	0,928208	0,32002	0,750341	-1,56924	2,16333	0,14852	0,464104	-0,78462	1,08167
3Q by 4L	-0,68893	0,928208	-0,74222	0,461572	-2,55522	1,17735	-0,34447	0,464104	-1,27761	0,58868
3Q by 4Q	-0,32955	0,803852	-0,40996	0,683656	-1,94580	1,28670	-0,16478	0,401926	-0,97290	0,64335

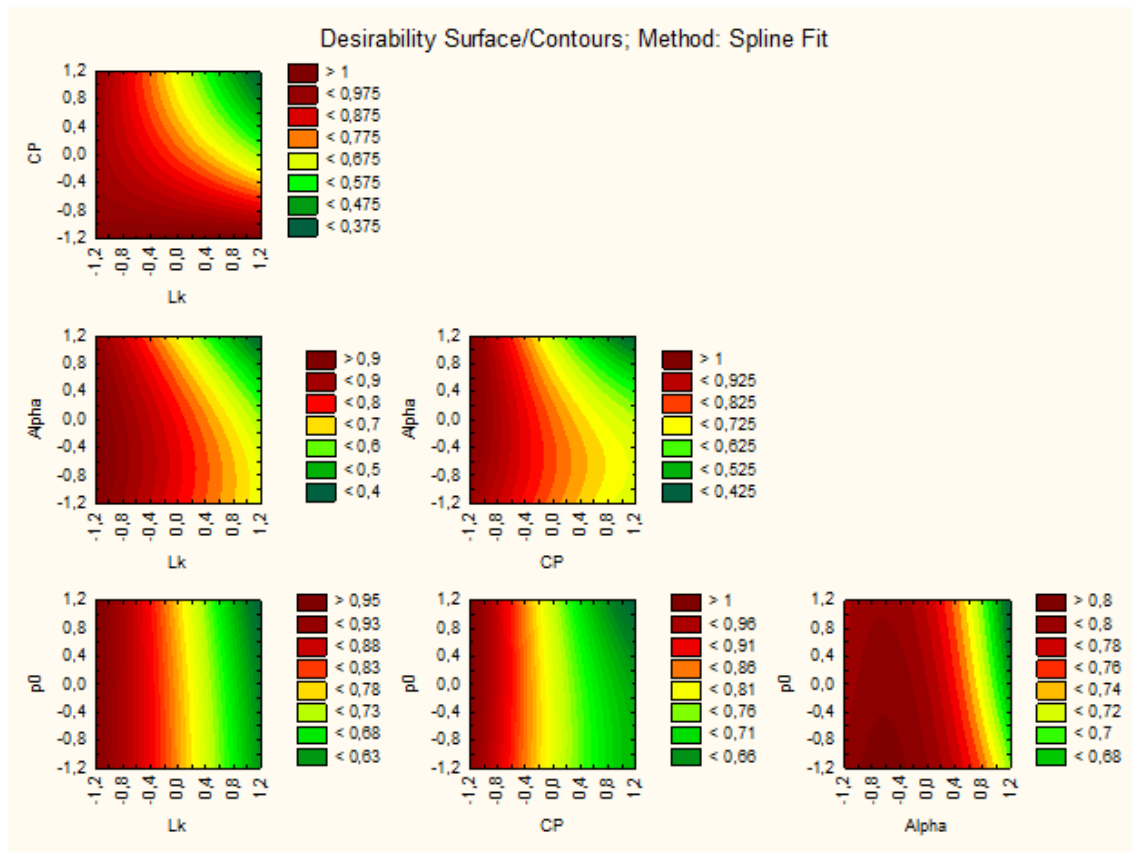


Figura .9: Gráfico de contorno das quatro variáveis relevantes, relativamente ao tempo, do modelo 3 do problema 1

.2 Appendix:Imagens

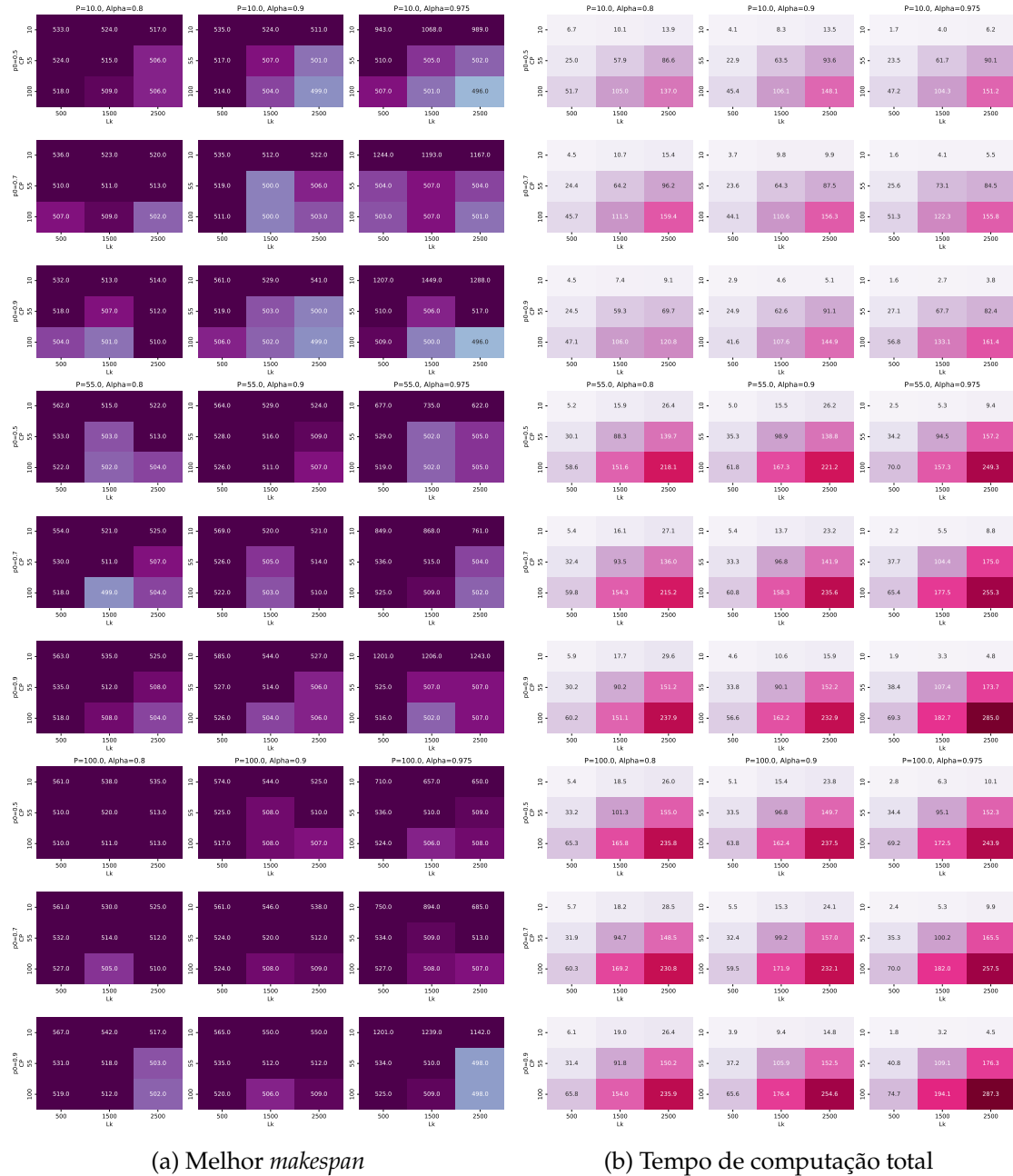


Figura .10: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 1.

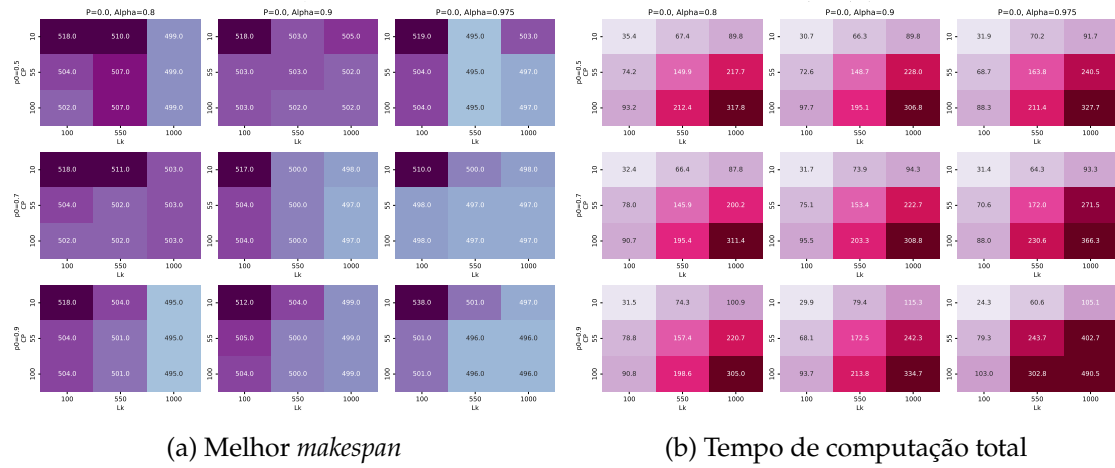


Figura .11: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 2.

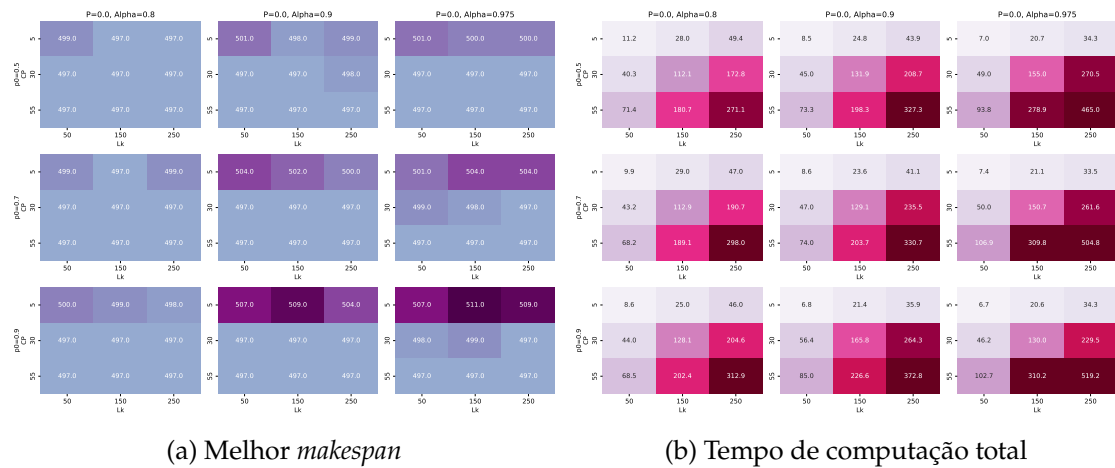


Figura .12: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 3 com *left shifting*.

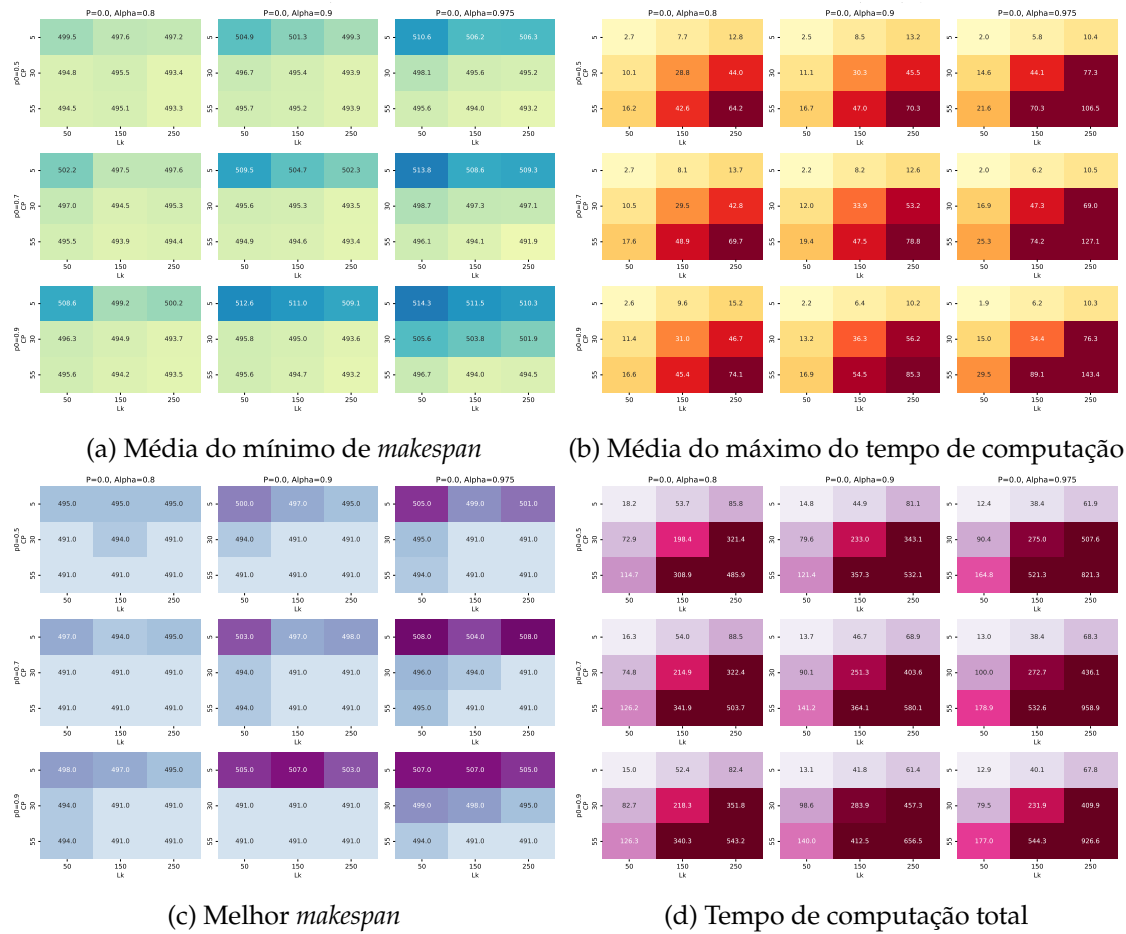


Figura .13: Impacto dos níveis das variáveis sobre o valor da função objetivo e do tempo computacional para o modelo 3 com *enhanced left shifting*.

