



Cristiano José Chainho Pereira

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Suporte IoT para Otimização de espaços físicos em ambiente Industrial

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Ricardo Luís Rosa Jardim Gonçalves, Professor Associado com Agregação, Departamento de Engenharia Eletrotécnica Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Coorientador: Carlos Manuel Melo Agostinho, Investigador, Centre of Technology and Systems - UNINOVA

Júri:

Presidente: Doutora Anabela Monteiro Gonçalves Pronto

Arguentes: Doutor João Francisco Alves Martins

Vogais: Doutor Carlos Manuel de Melo Agostinho

Suporte IoT para Otimização de espaços físicos em ambiente Industrial

Copyright © Cristiano José Chainho Pereira, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Aos meus pais e irmã

Agradecimentos

Em primeiro lugar não posso deixar de agradecer à minha família, eles sem duvida foram e são o meu suporte em todas as etapas da minha vida inclusive nesta que agora termina. No seu esforço para me apoiarem, incentivaram-me a ser mais e melhor pessoa. A eles um enorme obrigado por tudo.

Em segundo, quero agradecer ao Professor Doutor Ricardo Jardim Gonçalves e Doutor Carlos Agostinho pela orientação, e ajuda, para a concretização deste trabalho, bem como ao projeto C2NET por me permitir realizar este trabalho em contacto com este sector tão desafiante que é o sector industrial.

Um enorme obrigado à Raquel Melo pela paciência e auxilio prestados no decorrer não só deste trabalho, mas também de todo o meu percurso académico, sem duvida uma amiga, mais uma vez obrigado.

Quero agradecer também a todos os que me acompanharam nesta jornada que é vida académica. Pessoas como, o António Gonçalves, o Miguel Rodrigues, o Pedro Viegas, o Carlos Simão, o Fábio Oliveira, o Filipe Viegas, o Ali Saad e o Alexander Fernandes foram sem duvida pessoas que valeu a pena partilhar momentos, sorrisos, piadas, desgostos, horas de estudo e tudo mais que este caminho nos proporciona.

A todos muito Obrigado.

Resumo

A globalização e a conseqüente evolução dos mercados conduziram a um aumento da procura e oferta de produtos. Estes factos refletiram-se dando origem à conhecida sociedade de consumo presente nos dias de hoje. Como conseqüência dos factos enumerados advém a crescente necessidade de transporte e armazenamento de produtos, o que se traduz em maiores custos para as empresas e conseqüentemente para o consumidor final.

Assim, a necessidade de melhorar e otimizar estes processos torna-se uma questão pertinente, que desde cedo ganhou o seu lugar junto da comunidade científica. Melhorar o processo de otimização de espaço em veículos de transporte ou armazéns, revelou-se uma solução eficaz de forma a reduzir custos.

Portanto, torna-se imperativo criar um sistema que permite a otimização destes processos de uma forma automática. Assim esta tese propõe a criação de um sistema que permita a otimização do espaço de uma forma automática e que tenha em consideração aspetos relevantes, minimizando custos e maximizando recursos úteis. Para tal esta solução apresenta uma forma melhorada para a classificação dos problemas de otimização de espaço, uma rede de sensores capaz de adquirir a informação referente aos produtos a acomodar, e por fim, toda a arquitetura necessária à implementação deste sistema em ambiente industrial.

Palavras-Chave: Sistema Inteligente, Internet das Coisas, Classificação, Problemas de Otimização

Abstract

Globalization and the consequent evolution of the markets led to an increase in demand and supply of products. These facts are reflected in giving rise to this known consumer society today. As a result of the enumerated facts stemmed the growing need for transport and storage of products, which translates into higher costs for businesses and, consequently, to the final consumer.

Thus, the need to improve and optimize processes becomes a pertinent question that early earned its place among the scientific community. Improve space optimization process in transport vehicles or warehouses, has proved to be an effective solution to reduce costs.

Therefore, it is imperative to create a system that allows the optimization of these processes in an automated way. So this thesis proposes the creation of a system that allows the optimization of an automatic space, taking into account relevant aspects, minimizing costs and maximizing useful resources. To this solution has such a shape for improved classification of space optimization problems, a sensor network capable of acquiring information concerning sleep products, and finally, the whole architecture required to implement this system in an industrial environment.

Keywords: Intelligent System, Internet of Things, Classification, Optimization Problems

Índice

AGRADECIMENTOS	V
RESUMO	VII
ABSTRACT	IX
ÍNDICE DE FIGURAS	XV
ÍNDICE DE TABELAS	XVII
LISTA DE ACRÓNIMOS	XIX
CAPÍTULO 1 - INTRODUÇÃO	1
1.1. METODOLOGIA DE TRABALHO	3
1.1.1. <i>Caracterização do problema</i>	4
1.1.2. <i>Investigação do Estado da Arte</i>	4
1.1.3. <i>Formulação da Hipótese</i>	4
1.1.4. <i>Preparação de uma Experiência</i>	4
1.1.5. <i>Teste da Hipótese</i>	4
1.1.6. <i>Validação da Hipótese</i>	4
1.1.7. <i>Divulgação dos Resultados</i>	5
1.2. ORGANIZAÇÃO DO DOCUMENTO	5
CAPÍTULO 2 - PESQUISA E ESTADO DA ARTE	7
2.1. PROBLEMAS DE CORTE E ACOMODAÇÃO	7
2.1.1. <i>Identical Item Packing Problem</i>	9
2.1.2. <i>Placement Problem</i>	9
2.1.3. <i>Knapsack Problem</i>	10

2.1.4. <i>Open Dimension Problem</i>	11
2.1.5. <i>Cutting Stock Problem</i>	11
2.1.6. <i>Bin Packing Problem</i>	12
2.1.7. <i>Análise – Problemas de corte e acomodação</i>	13
2.2. ESTRATÉGIAS PARA AQUISIÇÃO AUTOMÁTICA DE DADOS.....	14
2.2.1. <i>IoT-A</i>	17
2.2.2. <i>OpenIoT</i>	18
2.2.3. <i>FIWARE IoT-Stack</i>	20
2.2.4. <i>C2NET – IoT Hub</i>	21
2.2.5. <i>Análise – Estratégias para a aquisição Automática de Dados</i>	22
CAPÍTULO 3 – ACOMODAÇÃO POR PRIORIDADES SUPORTADA POR IOT.....	25
3.1. CARACTERIZAÇÃO DO PROBLEMA POR PRIORIDADES	25
3.2. SISTEMA DE OTIMIZAÇÃO INTELIGENTE.....	27
3.2.1. <i>Interface de Utilizador</i>	28
3.2.2. <i>Classificador de Cenários</i>	28
3.2.3. <i>Optimizador de Espaço</i>	28
3.2.4. <i>Base de Dados</i>	29
3.2.5. <i>Mediador IoT</i>	29
3.2.6. <i>Rede de Sensores</i>	29
3.2.7. <i>Funcionalidade Integrada</i>	29
3.3. PROVA DE CONCEITO - IMPLEMENTAÇÃO	30
3.3.1. <i>Interface de Utilizador:</i>	31
3.3.2. <i>Classificador de Cenários:</i>	32
3.3.3. <i>Optimizador de Espaço</i>	34
3.3.4. <i>Base de Dados</i>	36
3.3.5. <i>Mediador IoT</i>	37
3.3.6. <i>Rede de Sensores</i>	37
CAPÍTULO 4 - TESTES E VALIDAÇÃO	39
4.1. METODOLOGIA DE TESTE.....	39
4.2. DEFINIÇÃO E EXECUÇÃO DE TESTES	41
4.2.1. <i>Definição do Teste: Aquisição de dados</i>	41
4.2.2. <i>Execução do Teste</i>	42
4.2.3. <i>Definição do Teste: Otimização de Espaço</i>	43
4.2.4. <i>Execução do Teste</i>	44

4.3. ANÁLISE DOS TESTES EFETUADOS	45
4.4. VALIDAÇÃO DO CENÁRIO INDUSTRIAL	45
CAPÍTULO 5 – CONCLUSÕES E TRABALHO FUTURO	47
5.1. TRABALHO FUTURO	49
CAPÍTULO 6 - BIBLIOGRAFIA.....	51

Índice de Figuras

Figura 1.1 - Método Científico	3
Figura 2.1 – Tipos básicos de problemas de corte e acomodação, baseado em (Wäscher, Haußner, & Schumann, 2007)	8
Figura 2.2 - Tipos de Variedade de itens.....	8
Figura 2.3 - <i>Identical Item Packing Problem</i> (IIPP).....	9
Figura 2.4 - <i>Single Large Object Placement Problem</i> (SLOPP).....	10
Figura 2.5 – <i>Single Knapsack Problem</i> (SKP).....	10
Figura 2.6 - <i>Open Dimension Problem</i> (ODP).....	11
Figura 2.7 - <i>Single Stock-Size Cutting Stock Problem</i> (SSSCSP).....	12
Figura 2.8 - <i>Single Bin Size Bin Packing Problem</i> (SBSBPP).....	12
Figura 2.9 - Evolução do IoT (S. Li et al., 2015)	16
Figura 2.10 - Modelo Funcional IoT-A (Bauer et al., 2013).	17
Figura 2.11 – Arquitetura <i>OpenIoT</i> (Medvedev et al., 2015).	19
Figura 2.12 - Arquitetura <i>FIWARE IoT Stack</i> (“FIWARE Components - FIWARE-IOT-Stack,” 2015).	20
Figura 2.13 - C2NET - <i>IoT Hub</i> (Partners C2NET, 2015).....	22
Figura 3.1 – Árvore de classificação baseada em prioridades.....	26
Figura 3.2 – Arquitetura do Sistema de Otimização Inteligente.	28

Figura 3.3 – Interação de módulos no processo de otimização.	30
Figura 3.4 - Interface de Utilizador.....	31
Figura 3.5 - Tabela de Itens.	32
Figura 3.6 - Coluna de índice de prioridade.	32
Figura 3.7 - Fluxograma da decisão no Classificador	33
Figura 3.8 - Vetor de Classes	34
Figura 3.9 - Vetor de prioridades	34
Figura 3.10 - Vetor de Espaços disponíveis.	35
Figura 3.11 - Pseudo-código do Optimizador para o problema <i>Bin Packing Problem</i>	35
Figura 3.12 - Tabela de Itens.	36
Figura 3.13 – a) Arduino Mega, b) Sensor ultrassónico HC-SR04, c) Shield NFC/RFID NP532, d) Mifare Classic Card e e) Rede de Sensores Final.....	38
Figura 4.1 - Processo de Teste.....	39
Figura 4.2 - Cenário Industrial - Armazém.....	46

Índice de Tabelas

Tabela 2.1 - Problemas específicos com maximização da saída, baseada em (Wäscher et al., 2007).	13
Tabela 2.2 - Problemas específicos com minimização da entrada, baseada em (Wäscher et al., 2007)	13
Tabela 2.3 - Comparativo de Soluções IoT.	23
Tabela 3.1 – Tabela resumo da classificação de Problemas.	27
Tabela 4.1 - Exemplo de Tabela de teste baseada em TTCN	40
Tabela 4.2 - Tabela exemplo caso de teste.....	41
Tabela 4.3 – Definição do Teste: Aquisição de Dados.....	42
Tabela 4.4 - Execução do Teste: Aquisição de Dados.....	42
Tabela 4.5 - Definição do Teste: Otimização de espaço	43
Tabela 4.6 - Execução do Teste: Otimização de espaço	44

Lista de Acrónimos

IIPP – *Identical Item Packing Problem*

SLOPP – *Single Large Object Packing Problem*

SKP – *Single Knapsack Problem*

ODP – *Open Dimension Problem*

CSP – *Cutting Stock Problem*

SSSCSP – *Single Stock Size Cutting Stock Problem*

BPP – *Bin Packing Problem*

SBSBPP – *Single Bin Size Bin Packing Problem*

MILOPP – *Multiple Identical Large Object Placement Problem*

MHLOPP – *Multiple Heterogeneous Large Object Placement Problem*

MIKP – *Multiple Identical Knapsack Problem*

MHKP – *Multiple Heterogeneous Knapsack Problem*

MSSCSP – *Multiple Stock Size Cutting Stock Problem*

RCSP – *Residual Cutting Stock Problem*

MBSBPP – *Multiple Bin Size Bin Packing Problem*

RBPP – *Residual Bin Packing Problem*

CPS – *Cyber Physical Systems*

IoT – *Internet-of-Things*

RFID – *Radio-Frequency IDentification*

WSN – *Wireless Sensor Network*

NFC – *Near Field Comunication*

IoT-A – *Internet-of-Things Architecture*

FG – *Functionality Groups*

VE – *Virtual Entity*

SSN – *Semantics Sensor Network*

NGSI – *Next Generation Services Interface*

STH – *Short Term Historic*

MQTT – *Message Queuing Telemetry Transport*

XMPP – *Extensible Messaging and Presence Protocol*

DDS – *Data Distribution Service*

AMQP – *Advanced Message Queuing Protocol*

IU – *Interface de Utilizador*

BD – *Base de Dados*

HTML – *Hiper Text Markup Language*

UID – *Unique IDentifier*

REST – *Representational State Transfer*

JSON – *JavaScript Object Notation*

PME – *Pequena e Média Empresa*

TTCN – *Testing and Test Control Notation*

Capítulo 1 - Introdução

O transporte e armazenamento de coisas, ou seja, utensílios, ferramentas ou alimentos, tem sido desde o início dos tempos um motivo de preocupação que envolve a Humanidade. No passado existia uma necessidade urgente de guardar coisas, alimentos ou bens de forma a que permitir a subsistência no dia-a-dia ou em períodos de escassez causados por razões naturais ou pelo próprio Homem. Acompanhando a evolução das civilizações, e o surgimento de aglomerados populacionais estas questões ganharam uma maior força. A necessidade de transportar e armazenar bens para as comunidades revelou-se assim indispensável de forma a responder às necessidades impostas pelo consumo, nomeadamente o desejo obter os bens de forma mais rápida e em maior quantidade, e as quais se traduziram na sociedade de consumo à qual assistimos hoje em dia.

A globalização criou “pontes” entre todos os pontos do mundo, afetando vários aspetos da sociedade moderna, sendo um deles o comércio internacional. As importações e exportações são um reflexo da quantidade de produtos que requerem transporte e armazenamento em por todo o mundo, levando à normalidade que é nos dias de hoje ter acesso a produtos importados.

Assim as necessidades impostas pelo consumo e a globalização obrigaram ao melhoramento dos processos industriais e dos seus mercados. De forma a “alimentar” o consumo crescente da sociedade, os processos industriais tornaram-se mais rápidos e eficientes e os mercados mais criativos fornecendo novos e melhores produtos ao consumidor final. Contudo estas necessidades também vieram desafiar os processos de transporte e armazenamento destes produtos. Sendo estes os responsáveis por estabelecer a comunicação entre as indústrias e mercados com o consumidor final, é necessário que as indústrias produtoras e transportadoras tenham uma resposta mais rápida e eficiente.

No caso do transporte, é necessário garantir que os bens chegam aos seus destinos devidos e de forma adequada, reconhecendo a importância das necessidades específicas de cada setor ou pessoa. No caso do armazenamento de produtos, a necessidade de melhoramento prende-

se no objetivo de alcançar uma melhor organização, tendo em conta a sua data de validade ou até a ordem de saída requerida, entre outros.

Contudo, esses problemas relacionados com otimização e organização de espaço vão permitir que mais bens sejam armazenados ou transportados com maior eficiência, ou seja, transportados num menor espaço de tempo ou armazenados em maior quantidade. O carregamento de um camião, carro ou carrinha pode ser efetuado de uma forma simples, mas na área da logística, espaço pode ser traduzido em dinheiro. Isso é observável quando se pensa que espaço desperdiçado num transporte pode significar que alguns produtos ou materiais poderiam ser transportados mais cedo, o que se poderia traduzir em menos viagens e consequentemente em menos gastos nessas viagens. Num cenário de armazenamento, esse mesmo espaço desperdiçado, ou uma má otimização ou organização do espaço disponível, pode ser traduzido em produtos que não estão disponíveis em caso de necessidade, arrumação de produtos de forma inapropriada sendo que estas falhas podem representar diminuição de lucro para a empresa e falha na manutenção e gestão de stock (Pedruzzi, Paulo, Nunes, Rosa, & Arpini, 2016; Schwerin & Wäscher, 1997).

A área da logística não é a única com esta preocupação. Na área da indústria da manufatura os problemas anteriores também podem ser identificados dada a necessidade de transportar e armazenar, tanto as matérias primas como os produtos acabados. Tal como na logística os custos derivados do mau armazenamento podem crescer descontroladamente (Silva, Araujo, & Poldi, 2015; Viegas, Vieira, Henriques, & Sousa, 2015).

Tendo em conta este tipo de problemas de otimização de espaços físicos a comunidade científica sentiu necessidade de procurar soluções por forma a minimizar os desperdícios, sendo o primeiro estudo datado de 1970 (Haims & Freeman, 1970). Estas soluções para a otimização procuram solucionar cenários tais como: necessidade de selecionar os produtos a acomodar de forma a obter o melhor aproveitamento do espaço ou quando se pretende minimizar o espaço necessário para acomodação dos produtos.

No entanto, para a aplicação destas técnicas é necessário que os dados relevantes, tais como as dimensões dos produtos, fragilidade, data de saída, peso, lucro ou custo, sejam recolhidos e estejam disponíveis para serem introduzidos dinamicamente no sistema de otimização, processos esses que normalmente ocorrem manualmente sendo executados por um operador. O processo de aquisição de dados pode ser demorado e trabalhoso, sem menosprezar o facto de este estar sujeito ao erro humano. De notar que a quantidade de dados que é necessário recolher de forma a corresponder aos cenários acima descritos, e que permitirá solucionar estes problemas é elevada.

É urgente automatizar o processo para que se consiga proporcionar uma maior capacidade de resposta para o processo de otimização do espaço, permitindo que o armazenamento seja mais eficiente e com menos esforço. Contudo, ainda não são encontradas soluções configuráveis pelo

utilizador que sejam capazes de determinar a melhor solução de otimização suportada por uma aquisição automática de dados.

Posto isto torna-se pertinente achar uma resposta para a seguinte questão:

Como criar um sistema dinâmico que permita a otimização automática de espaços físicos fechados mediante as condicionantes variáveis dos ambientes industriais?

Na qual a seguinte hipótese se coloca:

Se for possível recolher de forma automática a informação geométrica referente aos itens a acomodar, e, descobrir qual a forma mais eficiente de os arrumar dentro do espaço disponível consoante características definidas pelo utilizador, então, será possível criar um sistema dinâmico que optimize automaticamente o espaço disponível.

1.1. Metodologia de Trabalho

Para a realização deste trabalho analisou-se qual seria o melhor método científico a seguir, sendo que foi determinado que a melhor metodologia de trabalho a ser utilizada é a baseada no Método Científico (Schafersman, 1997) que é constituída por os sete passos apresentados na figura abaixo (Figura 1.1).

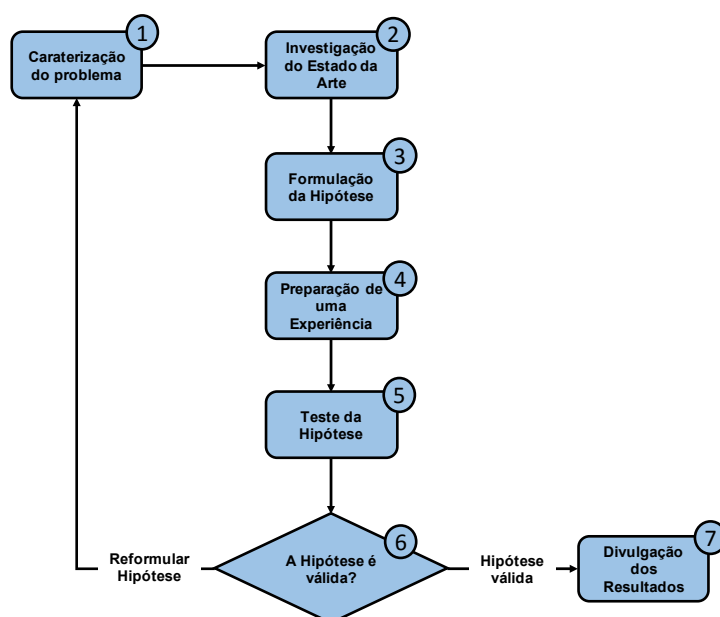


Figura 1.1 - Método Científico

1.1.1. Caracterização do problema

Neste primeiro passo do método científico é onde se identifica o problema e as suas características, terminando com a Pergunta de Investigação. O problema identificado neste trabalho é como criar um sistema que permita a otimização automática de espaços físicos fechados.

1.1.2. Investigação do Estado da Arte

Este passo dedica-se à investigação do trabalho relacionado já realizado, relevante para o problema em estudo. Neste trabalho, este passo foca-se na investigação de trabalhos realizados nas áreas da otimização de espaços e da aquisição automática de dados.

1.1.3. Formulação da Hipótese

Tendo como base a investigação realizada a hipótese deve clarificar, especificar e explicar a caracterização do problema, de forma a que este possa ser resolvido. Neste trabalho a hipótese foca-se na determinação de qual a forma mais eficiente de acomodar os itens e na forma de adquirir de forma automática a informação dos itens a acomodar.

1.1.4. Preparação de uma Experiência

Neste trabalho a experiência consiste na implementação e integração de um sistema para a aquisição de informação geométrica com uma plataforma de auxílio ao utilizador que permita a otimização de espaço.

1.1.5. Teste da Hipótese

Neste passo primeiro é necessário definir o teste de acordo com as características do problema e a hipótese formulada. É necessário avaliar a resposta da plataforma de forma a avaliar a hipótese proposta. Para tal a plataforma deve ser testada.

1.1.6. Validação da Hipótese

Após a validação dos resultados provenientes dos testes, a verificação da validade da hipótese proposta tem de ser feita. A validação desta necessita de ter em conta as características do problema. Os resultados podem potenciar a hipótese ou destruir todo o feito até este ponto, o que levará à necessidade de formular uma nova hipótese e refazer todo o trabalho até a passo três.

1.1.7. Divulgação dos Resultados

Após resultados positivos é possível considerar o trabalho válido e definir recomendações para trabalhos futuros. Os resultados devem resultar numa contribuição para a comunidade científica e deve ser finalizado com uma dissertação sobre a hipótese.

1.2. Organização do Documento

Alem deste capítulo, que já foi apresentado, esta dissertação é constituída por outros quatro capítulos, as características de cada um deles serão apresentadas em seguida:

Pesquisa e Estado da Arte: O segundo capítulo apresenta os elementos estudados de trabalhos relacionados, que são, problemas de otimização de espaço e estratégias para a aquisição automática de dados. Em cada um destes pontos os trabalhos relacionados são apresentados e analisados de forma a concluir a sua viabilidade para a solução deste problema.

Solução: No terceiro capítulo é apresentado uma nova forma de caracterização dos problemas de otimização de espaço, bem como uma arquitetura que conciliando os dois tópicos alvos de investigação mencionados no capítulo anterior se propõem que resolva a questão na qual assenta esta dissertação. Este capítulo conta também com a descrição da implementação realizada para materializar a arquitetura apresentada

Teste e Validação: O terceiro capítulo apresenta o tipo de teste utilizada para validar a hipótese. Assim é apresentada a metodologia do teste adotado para testar a hipótese. Também é apresentado neste capítulo o cenário de prova de conceito desta dissertação, sendo que este capítulo termina com a averiguação de se a solução apresentada alcança os objetivos pretendidos para este trabalho.

Conclusões e Trabalho futuro: O quinto capítulo apresenta então um sumário da dissertação, destacando os aspetos mais importantes do trabalho realizado. Este capítulo também leva em conta os resultados obtidos no processo de teste e aponta uma direção a ser tomada em pesquisas futuras.

Capítulo 2 - Pesquisa e Estado da Arte

Para sustentar a hipótese apresentada, foi efetuada uma pesquisa focada em duas partes como se apresenta seguidamente.

A primeira parte tem como foco da investigação os problemas de otimização de espaço, estes abordam a ocupação mais eficiente de espaços fisicamente fechados conduzindo ao menor desperdício desse espaço. Ao longo deste trabalho os produtos que pretendem ser acomodados serão denominados por **itens**, já no caso do local onde será efetuada a acomodação este será definido como **espaço**.

A segunda parte da pesquisa efetuada diz respeito a estratégias de aquisição de dados de forma automática, no sentido de compreender quais poderão ser os meios utilizados para essa aquisição com menos interação humana, dando preferência a uma aquisição em tempo real das características dos itens e local de acomodação (espaço) podendo estas ser dimensões geométricas, peso ou fragilidade.

2.1. Problemas de Corte e Acomodação

De acordo com (Delorme, Iori, & Martello, 2016), os tipos básicos de problemas de corte e acomodação (Figura 2.1) podem ser caracterizados com base em dois critérios:

- Tipo de tarefa/missão: Este critério é definido como tendo duas respostas possíveis, maximização da saída ou minimização da entrada. O primeiro corresponde a um tipo de problema em que o número de espaços é limitado e o valor de itens a acomodar deve ser maximizado, ou seja, pretende-se que o número de itens acomodados resultante da otimização seja maximizado. O segundo corresponde a um tipo de problema onde os espaços disponíveis para a acomodação dos itens é satisfatório sendo necessário minimizar o número de espaços usados para essa acomodação, ou seja, pretende-se que o número de espaços utilizados para a acomodação seja minimizado.

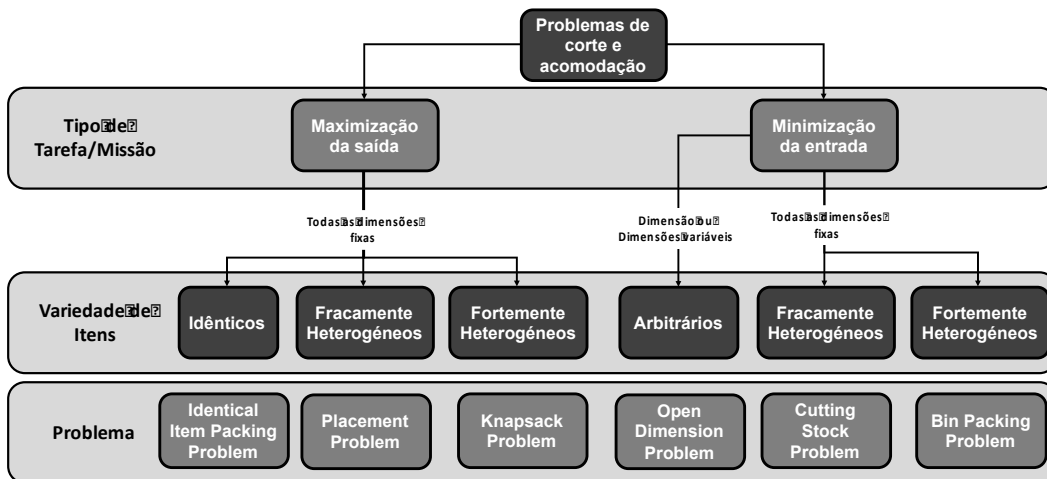


Figura 2.1 – Tipos básicos de problemas de corte e acomodação, baseado em (Wäscher, Haußner, & Schumann, 2007)

- Variedade de Itens: Aqui o critério relaciona-se com a forma e tamanho dos itens, sendo que neste critério os resultados possíveis são quatro e estão ilustrados na Figura 2.2. O primeiro é itens **Idênticos**, onde os itens têm o mesmo tamanho e forma. O segundo resultado possível é itens **Fracamente Heterogêneos** os quais podem ser agrupados em algumas classes de itens idênticos, de salientar que itens com diferentes orientações são considerados itens diferentes. O terceiro é itens **Fortemente Heterogêneos**, onde todos os itens são diferentes e por fim itens **Arbitrários**, onde os itens podem ser de qualquer um dos tipos mencionados anteriormente (Dyckhoff, 1990).

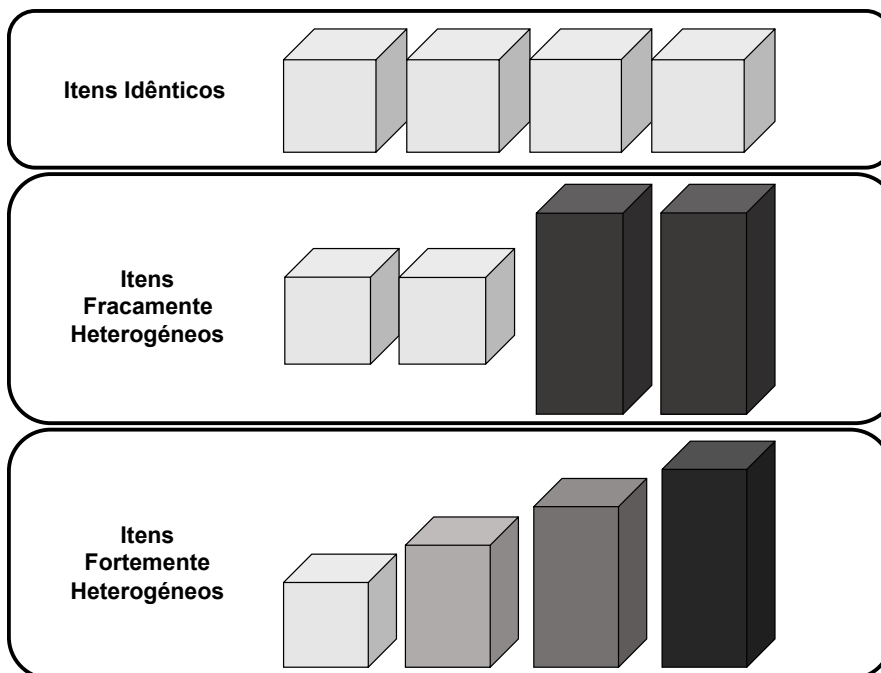


Figura 2.2 - Tipos de Variedade de itens.

Assim com esta caracterização seis tipos básicos de problemas são definidos. Eles são: *Identical Item Packing Problem*, *Placement Problem*, *Knapsack Problem*, *Open Dimension Problem*, *Cutting Stock Problem* e *Bin Packing Problem*.

2.1.1. Identical Item Packing Problem

Esta categoria de problemas é caracterizada por itens idênticos e maximização da saída como descrito na figura anterior. Tem como foco uma otimização de espaço que permita, uma maximização de espaço utilizado para a acomodação dos itens. Embora neste tipo de problemas os itens sejam idênticos e por isso não sejam primeiramente selecionados (Figura 2.3), neste caso a otimização irá revelar quantos itens iguais podem ser acomodados no espaço disponível, sendo assim este tipo de problemas podem ser resumidos a um problema de combinação geométrica de itens idênticos (Wäscher et al., 2007).

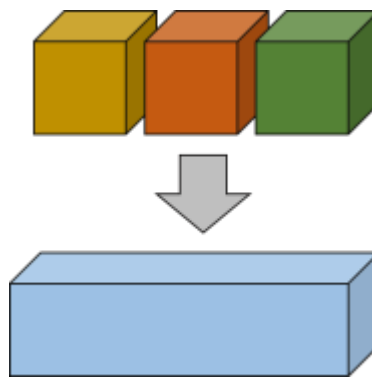


Figura 2.3 - *Identical Item Packing Problem* (IIPP).

2.1.2. Placement Problem

O *Placement Problem* é apresentado como uma categoria de problemas caracterizados por itens fracamente heterogêneos para serem acomodados num determinado conjunto de objetos disponíveis (Wäscher et al., 2007). Neste tipo de problemas, o valor e o número de itens devem ser maximizados.

Contudo, esta categoria de problema apresenta três problemas específicos, sendo eles: *Single Large Object Placement Problem* (Figura 2.4), *Multiple Identical Large Object Placement Problem* and *Multiple Heterogeneous Large Object Placement Problem*. Estes são resultado da disponibilidade e variedade de espaços para a acomodação (Wäscher et al., 2007), similarmente ao que acontece na tipologia de Dyckhoff (Dyckhoff, 1990).

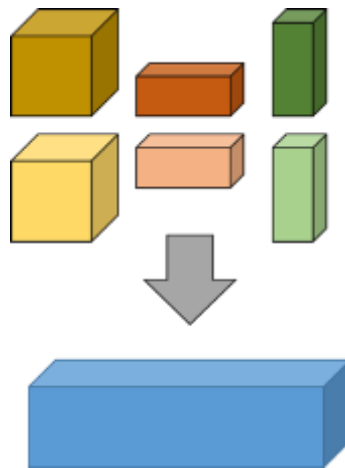


Figura 2.4 - *Single Large Object Placement Problem (SLOPP)*.

2.1.3. *Knapsack Problem*

De acordo com (Delorme et al., 2016) e (Wäscher et al., 2007) a categoria de problemas *Knapsack*, ou problema da mochila, é caracterizado por forte heterogeneidade dos itens que devem ser acomodados dentro do espaço (*Knapsack*). Neste tipo de problema os espaços disponíveis são limitados e por isso nem todos os itens podem ser acomodados. A decisão de qual o item deve ser acomodado é feita com base no valor (lucro) de cada item e com base espaço livre restante. Então este tipo de problema é focado em maximizar o lucro e o numero de itens no interior do espaço. Contudo, o conjunto de objetos disponíveis, à imagem do que foi explicado anteriormente origina três problemas específicos que são: *Single Knapsack Problem* (Figura 2.5), *Multiple Identical Knapsack Problem* e *Multiple Heterogeneous Knapsack Problem*.

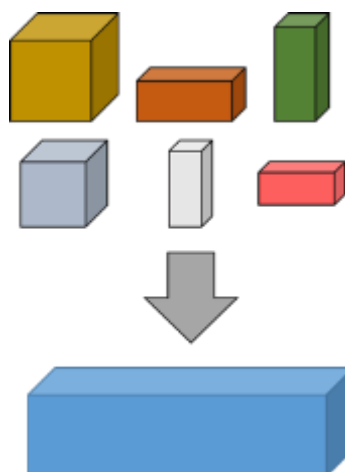


Figura 2.5 – *Single Knapsack Problem (SKP)*.

2.1.4. Open Dimension Problem

Esta categoria apresenta um tipo de problema onde o espaço tem pelo menos uma dimensão variável. Em outras palavras, pode ser comparado a uma caixa (espaço) sem tampa, resultando que essa caixa ganhe uma dimensão infinita, ou seja, sem limite (Figura 2.6). Nesta categoria todos os itens necessitam de ser acomodados no mesmo objeto (Wäscher et al., 2007) e (Baker, Coffman, Jr., & Rivest, 1980), e por isso o problema envolve a decisão de onde essa dimensão deve ser fixada, ou seja, qual a menor medida para essa variação de forma a permitir a acomodação de todos os itens. Resultando assim que nesta categoria de problemas o objetivo seja minimizar essa ou essas dimensões variáveis.

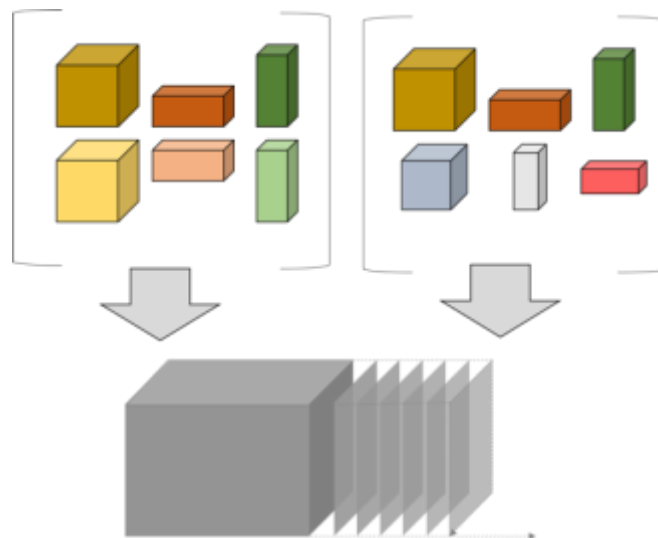


Figura 2.6 - Open Dimension Problem (ODP).

2.1.5. Cutting Stock Problem

A categoria de problemas *Cutting Stock Problem* (CSP), é caracterizada por uma variedade de itens fracamente heterogênicos e por uma seleção desses itens. Esta categoria preocupa-se com a minimização do número de espaços, minimização do custo associado aos espaços ou o tamanho total dos espaços utilizados para a acomodação de todos os itens (Delorme et al., 2016) e (Wäscher et al., 2007).

Então tendo em conta que os espaços disponíveis podem ser idênticos, fracos ou fortemente heterogêneos, isto leva a que como anteriormente este problema tenha três problemas específicos, *Single Stock-Size Cutting Stock Problem* (Figura 2.7), *Multiple Stock-Size Cutting Stock Problem* e *Residual Cutting Stock Problem*.

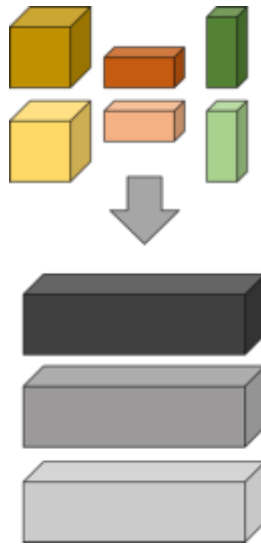


Figura 2.7 - *Single Stock-Size Cutting Stock Problem (SSSCSP)*.

2.1.6. *Bin Packing Problem*

De acordo com (Wäscher et al., 2007) e (Delorme et al., 2016), a categoria *Bin Packing Problem* (BPP) é caracterizada por um conjunto de itens fortemente heterogêneos, onde todo este conjunto de itens deve ser acomodado dentro de um conjunto de espaços (*Bins*) disponíveis que também podem ser Idênticos, forte ou fracamente heterogêneos resultando nos problemas específicos, *Single Bin Size Bin Packing Problem*, *Multi Bin Size Bin Packing Problem* e *Residual Bin Packing Problem*. O objetivo neste tipo de problema é minimizar o número, custo ou tamanho total dos objetos necessários para a acomodação dos itens.

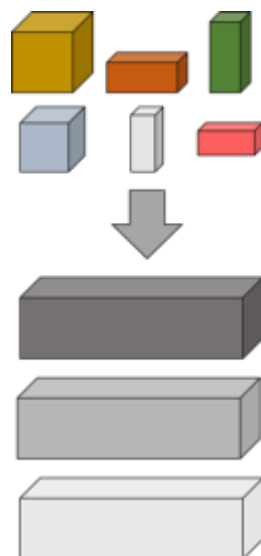


Figura 2.8 - *Single Bin Size Bin Packing Problem (SBSBPP)*.

2.1.7. Análise – Problemas de corte e acomodação

Assim por fim e de forma a ser possível resumir e identificar todos os problemas que respondem a um cenário de otimização de espaço, são apresentadas duas tabelas correspondentes aos problemas de maximização da saída (Tabela 2.1) e minimização da entrada (Tabela 2.2).

Tabela 2.1 - Problemas específicos com maximização da saída, baseada em (Wäscher et al., 2007).

Variedade de espaços \ Variedade de Itens		Variedade de Itens		
		Idênticos	Fracamente Heterogêneos	Fortemente Heterogêneos
Todas as dimensões fixas	Um	Identical Items Packing Problem (IIPP)	Single Large Object Placement Problem (SLOPP)	Single Knapsack Problem (SKP)
	Idênticos		Multiple Identical Large Object Placement Problem (MILOPP)	Multiple Identical Knapsack Problem (MIKP)
	Heterogêneos		Multiple Heterogeneous Large Object Placement Problem (MHLOPP)	Multiple Heterogeneous Knapsack Problem (MHKP)

Tabela 2.2 - Problemas específicos com minimização da entrada, baseada em (Wäscher et al., 2007)

Variedade de espaços \ Variedade de Itens		Variedade de Itens	
		Fracamente Heterogêneos	Fortemente Heterogêneos
Todas as dimensões fixas	Um	Single Stock Size Cutting Stock Problem (SSSCSP)	Single Bin Size Bin Packing Problem (SBSBPP)
	Idênticos	Multi Stock Size Cutting Stock Problem (MSSCSP)	Multi Bin Size Bin Packing Problem (MBSBPP)
	Heterogêneos	Residual Cutting Stock Problem (RCSP)	Residual Bin Packing Problem (RBPP)
Dimensões Variáveis	Um	Open Dimension Problem (ODP)	

Na visão geral das categorias de problemas apresentadas e para fazer uma pequena análise de forma a recolher os pontos principais destas, observamos que todas apresentam preocupações com a otimização do espaço. No entanto, essa otimização de espaço pode ser complementada com outros fatores. Em casos como o *Knapsack* e *Placement Problem* é também atribuído para a otimização um valor a cada item de forma a se obter um maior valor dentro do espaço. O *Knapsack* também apresentou a capacidade para aplicação em situações onde o peso é relevante para a otimização tendo como propósito respeitar a capacidade máxima de peso do espaço deixando para segundo plano a otimização em termos dimensionais.

Os problemas *Bin Packing* e o *Cutting Stock* são descritos com as mesmas preocupações no que se refere a: minimização do número, custo ou tamanho total dos espaços usados, e, por definição nessas categorias é garantido que todos os itens são acomodados. Contudo, nos problemas específicos denominados como residuais (RCSP e RBPP), onde a preocupação se torna a seleção dos espaços que conduzem à minimização do espaço desperdiçado, observa-se que os objetivos deste tipo de problemas passam a ser alcançados não só pela boa acomodação dos itens, mas também selecionando o tamanho do espaço que mais se adapta aos itens a acomodar.

Outra questão que a caracterização dos problemas de corte e acomodação levanta é o facto de ser considerado como critério de avaliação determinar se os itens presentes no cenário podem ou não ser todos acomodados no espaço disponível. Este aspeto torna-se interessante quando se conclui que para determinar isso será necessário efetuar uma primeira acomodação, aumentando consideravelmente a dificuldade e duração temporal que este processo iria exigir.

Assim, após este estudo conclui-se que nenhum problema dos identificados consegue solucionar um cenário geral, o que se pode traduzir na afirmação: não foi identificada uma solução geral para a otimização de espaços. Também é possível concluir que a caracterização estudada poderia ser melhorada de forma a facilitar este processo.

2.2. Estratégias para aquisição automática de Dados

Um *Cyber-Physical System* (CPS), em português Sistema Cyber-Físico, é definido como um sistema com capacidade de computação, rede e integração com processos físicos. De acordo com (Baheti & Gill, 2011), os *Cyber-Physical Systems* têm a capacidade de interagir e expandir as capacidades do mundo físico através da computação.

A evolução da tecnologia tanto em termos de custo como funcionalidades levou a que, as aplicações de CPSs estejam presentes em muitos campos, tais como sistemas no sector automóvel e de transporte, sistemas médicos e de cuidados de saúde, casas e edifícios inteligentes, sistemas de redes, controle de processos industriais e robótica (Khaitan & McCalley, 2015).

Na área de sistemas médicos e cuidados de saúde, em (Mitchell & Chen, 2015) a metodologia proposta tem preocupação com a detecção de intrusão de dispositivos médicos não autorizados em CPSs médicos, e revelam bons resultados, proporcionando maior segurança para os pacientes e sistemas. Outro trabalho apresentado em (Sonntag, 2014) tem foco na interação entre *Cyber-Physical Systems* e utilizadores no contexto do programa de rastreio do cancro. O autor propôs uma aplicação médica que, combinada com dispositivos aplicados ao paciente oferecem ao médico um ótimo suporte para o rastreio da doença.

Ainda outro exemplo da aplicação para os sistemas CPS é apresentado em (Yan et al., 2012), onde os autores propõem uma arquitetura hierárquica para um sistema CPS para um veículo não tripulado com navegação através de uma rede de sensores sem fios. Os movimentos do veículo são estabelecidos com recurso a leituras na rede de sensores e usando um algoritmo *Particle Swarming Optimization*, em português otimização de enxame de partículas, melhorando a velocidade de comunicação com o veículo, e conseqüentemente, a sua resposta.

As redes de sensores sem fios estão intrinsecamente ligadas à evolução dos sistemas CPS. Dado que estas são compostas por dispositivos com baixo consumo de energia e baixo poder de comunicação, chamados nós de sensores que são constituídos por um sensor, um microcontrolador e um controlador de comunicações sem fios que têm a capacidade de adquirir diversos tipos de sinais. Tendo em conta que uma rede de sensores está operacional quando o número mínimo de nós permanecem ativos é possível a comunicação com a estação base. A operacionalidade destas redes é determinada pela capacidade da bateria do nó e consumo de energia do mesmo (Kopetz, 2011).

Quando estas redes de sensores sem fios se ligam à *Internet (World Wide Web)*, assistimos à materialização de um novo paradigma, *Internet-of-Things (IoT)*, ou em português Internet das Coisas. A expressão Internet of Things foi utilizada pela primeira vez por Kevin Ashton em 1999, de acordo com (Ashton, 2009). A frase foi utilizada numa apresentação para cativar a atenção do executivo para uma ideia nova, de RFID (*Radio-Frequency IDentification*). Contudo, a frase ganhou impacto e de acordo com (S. Li, Xu, & Zhao, 2015), a sua definição varia mas apesar disso é implícito que os objetos são identificáveis na rede e que todas as coisas (things) são capazes de trocar e processar dados.

Para a utilização de tecnologias baseadas em IoT é necessário preencher a falha entre o mundo físico, onde se situam estas tecnologias, e o mundo digital, onde se encontram os sistemas (CPS) que podem retirar maiores rendimentos destas tecnologias (S. Li et al., 2015). As tecnologias IoT são vastas, o código de barras, as redes de sensores sem fios (*Wireless Sensor Networks - WSNs*), o RFID, os Sensores Inteligentes, o NFC (*Near Field Communication*), e *Cloud Computing* são exemplos disso (Jiang et al., 2014; Kang, Pang, Xu, Ma, & Wang, 2014; Kataev, Bulysheva, Emelyanenko, & Emelyanenko, 2013; Q. Li et al., 2012; Ren et al., 2012; Tao, Cheng, Xu, Zhang, & Li, 2014; Tao, Zuo, Xu, & Zhang, 2014). Como ilustrado na Figura 2.9 tem sido evidente a sua evolução ao longo dos últimos anos.

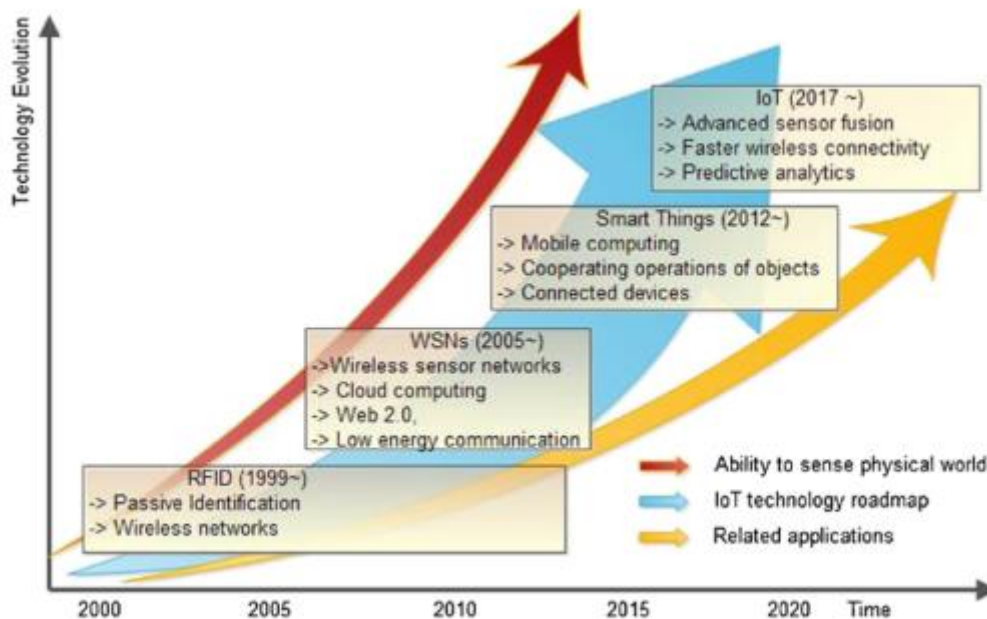


Figura 2.9 - Evolução do IoT (S. Li et al., 2015)

Como afirmado anteriormente, estes elementos IoT precisam ser capazes de partilhar e processar dados. A aquisição de dados através destas tecnologias é também importante para compreender o ambiente que os rodeia. Os CPSs podem ser usados para executar a aquisição de dados e ou deteção de dados, fazendo parte de um sistema de aquisição de dados capaz de coletar dados, processa-los localmente e introduzi-los nos sistemas ou aplicações inteligentes/centrais momento e forma corretos (Karnouskos, 2009). Estes sistemas de aquisição de dados apresentam preocupações tais como (Höller et al., 2014):

- A comunicação com dispositivos distribuídos: a comunicação pode ser feita através de ligações com e sem fios em resposta às exigências de aplicação e com protocolo de segurança.
- Natureza da aquisição: refere-se a quando os dados necessitam ser recolhidos, de forma contínua como em processos de monitoramento, baseado em eventos como em processos de alarme e assim por diante.
- Frequência: a frequência é determinada pelas necessidades da aplicação, em outras palavras, a frequência é o número de vezes por unidade de tempo que a aplicação requer a captura de informação.
- Aquisição vs. Geração: em sistemas de aquisição de dados, estes podem recolher dados e relatá-los ou, por outro lado, podem recolher dados e através da análise dos dados concluir estarem na presença de um determinado evento.

Este tipo de características devem ser foco de atenção nas camadas de ligação entre o mundo físico e o mundo digital, ou seja, entre a camada sensorial e a camada aplicacional, tornando-se

assim necessário compreender como acontece a ligação entre esses dois mundos e as soluções disponíveis para tal. Seguidamente são apresentadas algumas das tecnologias que já se encontram disponíveis ou em fase de protótipo, nomeadamente IoT-A, OpenIoT, FIWARE IoT Stack e C2NET – IoT HUB.

2.2.1. IoT-A

A IoT-A é um modelo de arquitetura de referência (ARM - *Architecture Reference Model*) desenvolvido com a intuito de alcançar uma solução global para sistemas IoT. Esta arquitetura é composta por um conjunto inicial de blocos que pretendem ser a base para as emergentes soluções IoT (“Internet of Things - Architecture — IOT-A: Internet of Things Architecture,” 2013).

O modelo funcional da IoT-A (Figura 2.10) apresenta a constituição desta arquitetura de referencia baseada nas suas funcionalidades, sendo composta por cinco grupos de funcionalidades (FG - *Functionality Groups*) longitudinais e dois grupos de funcionalidades transversais (Bauer, Boussard, Bui, & Carrez, 2013) e cada um deles será explicado seguidamente.

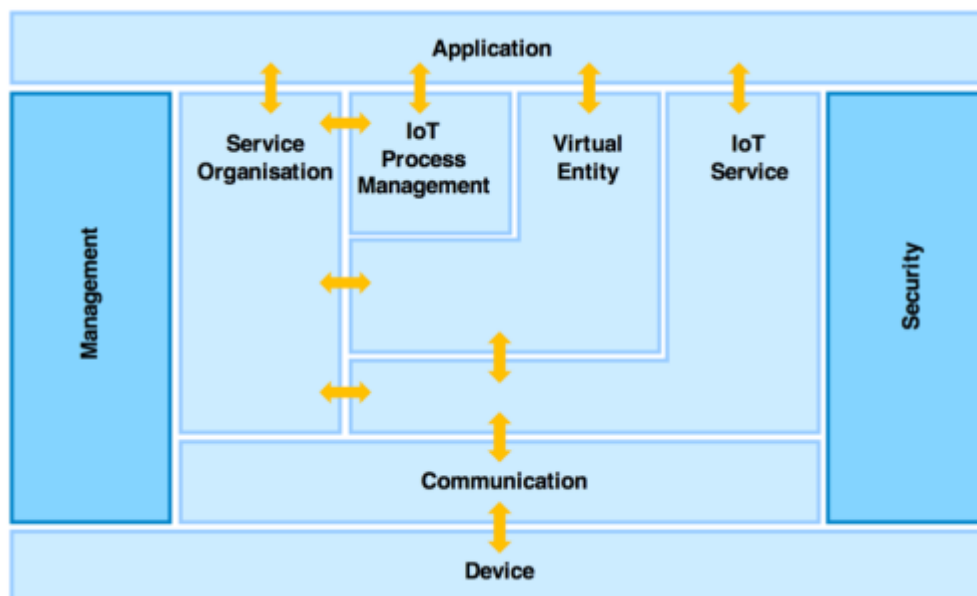


Figura 2.10 - Modelo Funcional IoT-A (Bauer et al., 2013).

- *IoT Process Management*: este FG é responsável pela integração conceptual do sistema de gestão de processos com a IoT-A, ou seja, possibilita a integração dos conceitos funcionais do mundo IoT com os dos processos tradicionais, possibilitando assim a utilização de subsistemas IoT evitando as despesas e custos associados a soluções IoT à medida.
- *Service Organization*: o grupo funcional de Organização de Serviços é visto como um grupo central, dado que este comunica com quatro outros grupos funcionais e é usado para orquestrar e compor serviços de diferentes níveis de abstração. Por exemplo é

responsável por ligar FGs tais como o *IoT Process Management* a serviços básicos associados a recursos através de entidades virtuais (VEs - *Virtual Entities*).

- *Virtual Entity*: este FG é responsável por interagir com sistema IoT e descobrir serviços que possam conceder informação à VE. É também responsável por gerir associações, descobrir e monitorizar a sua validade.
- *IoT Service*: aqui é onde permanecem os serviços IoT e é responsável por descobrir novos serviços e fazer a resolução do nome desses novos serviços IoT.
- *Communication*: assegura uma interface comum ao *IoT Service* FG. Assegurando também uma forma fácil de gerir o grande fluxo de informação associado.
- *Management*: este FG combina todas as funcionalidades para a gestão de sistemas IoT, e introduz benefícios tais como: redução de custo, lidação com problemas de utilização inesperada, tratamento de falhas e flexibilidade.
- *Security*: o *security* FG (grupo funcional segurança) é onde o registo inicial do cliente é tratado em termos de segurança. Este FG também é responsável por proteger os parâmetros privados dos utilizadores através da utilização de vários identificadores pseudoaleatórios para um único utilizador. Assim este FG garante a interação que é suposta entre pares e também a integridade e confidencialidade desta no primeiro contacto entre entidades.

2.2.2. OpenIoT

O *OpenIoT*, de acordo com o artigo (Medvedev, Zaslavsky, Khoruzhnikov, & Grudin, 2015), é uma plataforma *open source* que providencia interoperabilidade entre serviços IoT na *Cloud*. Para isso o *OpenIoT* usa a ontologia W3C *Semantics Sensor Network* (SSN), permitindo a representação de sensores físicos e virtuais. Além disso, a *OpenIoT* permite o desenvolvimento e implementação de aplicações de IoT quase sem programação usando para isso, as ferramentas visuais disponíveis. Outra característica desta plataforma é a capacidade de lidar com sensores móveis.

A arquitetura (Figura 2.11) desta solução é composta por sete elementos apresentados e explicados em seguida.

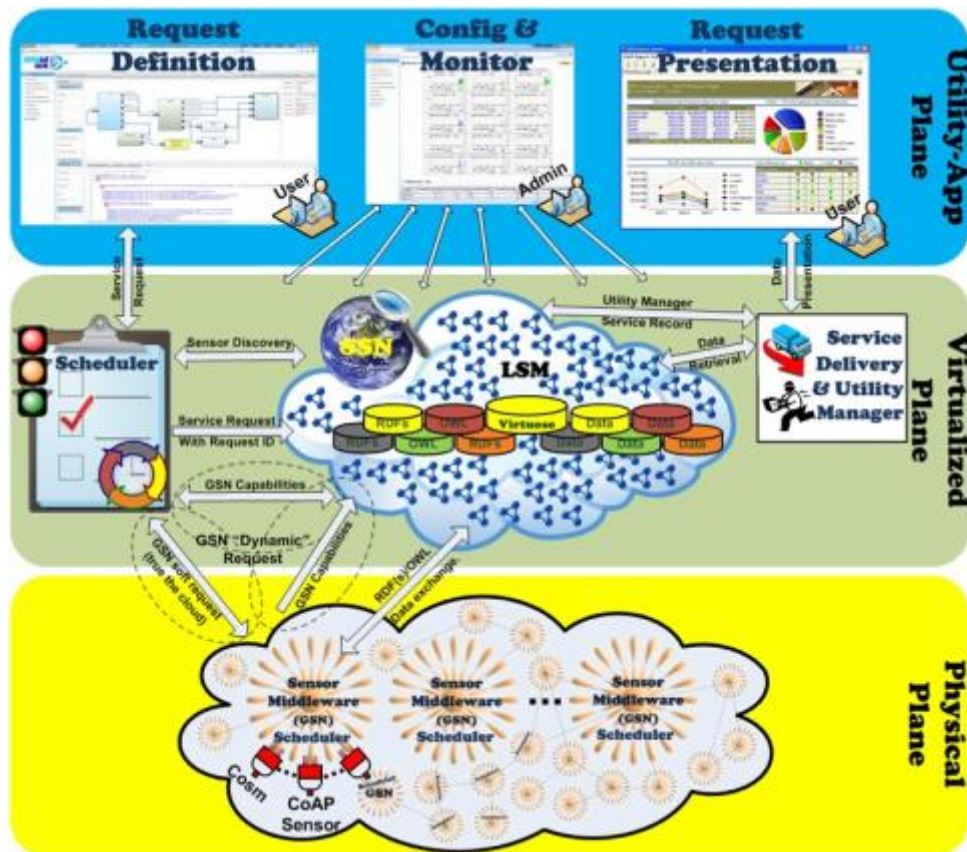


Figura 2.11 – Arquitetura *OpenIoT* (Medvedev et al., 2015).

- *Sensor Middleware*: este elemento é utilizado para adquirir, filtrar e combinar fluxos de dados provenientes de dispositivos físicos ou sensores virtuais. É também neste modulo que ocorre a integração com sensores moveis.
- *Cloud Data Storage*: é onde os fluxos de dados provenientes do *Sensor Middleware* são armazenados. É também responsável por armazenar os meta dados usados nas operações IoT.
- *Scheduler*: este elemento é responsável por processar as solicitações de serviços e fornecer a garantia de acesso aos recursos que são exigidas. Também identifica fluxos de dados e os sensores associados, que podem contribuir para um determinado serviço.
- *Service Delivery & Utility Manager*: combina os fluxos de dados com os sistemas *OpenIoT* para prestar o serviço necessário.
- *Request Definition*: o elemento *Request Definition* é responsável por identificar as especificações do serviço solicitado à plataforma *OpenIoT*. Também é responsável por seleccionar e compreender os serviços para responder a essas solicitações. Este elemento tem um suporte gráfico para o utilizador.
- *Request Presentation*: este componente é utilizado para efetuar a apresentação de realizações de serviços, utilizando para tal uma biblioteca apropriada para facilitar a apresentação do serviço.

- *Configuration and Monitoring*: este componente permite a gestão visual e configuração de serviços e sensores que são implantados usando a plataforma *OpenIoT*.

2.2.3. FIWARE IoT-Stack

FIWARE IoT Stack foi criado para permitir a conexão entre dispositivos e recetores de dados. Esta solução tem integrados todos os protocolos de dispositivos e métodos de conexão tornando assim possível a compreensão e interpretação da informação proveniente do dispositivo. Assim esta característica traduz-se no isolamento do dispositivo em relação ao processamento dos dados e também da complexidade da rede em termos de acesso e segurança (“FIWARE-IOT-Stack,” 2015a). Ou seja, o dispositivo poderá comunicar como quiser, pois, a tradução da sua “língua” será tratada pelo *FIWARE IoT Stack*.

Com isto o *FIWARE IoT Stack* oferece aos programadores uma forma simples de integração de sensores de dados, escalabilidade, programação modular aberta e baseada em standards, e API's (*Application Programing Interfaces*) independentes de dispositivos. Sendo assim fornecidas funcionalidades tais como a gestão de *tokens* necessários para a utilização das API's, gestão de dispositivos, envio de informação do dispositivo para a *Cloud*, análise de dados em tempo real entre outras (“FIWARE-IOT-Stack,” 2015b).

O *middleware Fiware IoT Stack* apresenta uma arquitetura composta por seis componentes (Figura 2.12). Esses são: *Device Backend Gateway*, *Context Broker*, *Short Term Historic*, *Connector Framework*, *Complex Event Processing* e *Identity Management*, que serão descritos em seguida (“FIWARE Components - FIWARE-IOT-Stack,” 2015).



Figura 2.12 - Arquitetura *FIWARE IoT Stack* (“FIWARE Components - FIWARE-IOT-Stack,” 2015).

- *Device Backend Gateway* (IDAS): usado para facilitar a gestão e integração de dispositivos. É responsável pela tradução dos protocolos utilizados pelos dispositivos para a linguagem padrão da plataforma, NGSI (*Next Generation Services Interface*) permitindo também enviar comandos para dispositivos (Bauer et al., 2010). Neste componente são suportados muitos protocolos IoT cada um pelo seu módulo específico (*Internet of Things Agent*), levando assim a uma separação dos dispositivos segundo o

seu protocolo de comunicação. Os protocolos suportados atualmente são: HTTP Ultralight 2.0, MQTT, OMALWM2M e Thinking Things Open (“Device Backend Gateway (IDAS) - FIWARE-IOT-Stack,” 2015).

- *Context Broker* (Orion): responsável por adquirir qual a informação relevante para aplicação. Através da utilização REST API é possível adquirir a informação no contexto das necessidades da aplicação, podendo estas informações ser adquiridas através de sistemas já existentes, aplicações moveis para utilizadores ou redes de sensores (“Context Broker (Orion) - FIWARE-IOT-Stack,” 2015).
- *Short Term Historic* (STH): este componente do *FIWARE IoT Stack* oferece aglomerados de informação temporais utilizando os dados registados através de do *Context Broker* (“Short Term Historic (STH) - FIWARE-IOT-Stack,” 2015).
- *Connector Framework* (Cygnus): responsável por conectar o *Context Broker* e um armazenamento de dados específico, tornando possível armazenar as informações provenientes do Orion em armazenamentos HDFS ou CKAN, ou então numa base de dados MySQL (“Connector Framework (Cygnus) - FIWARE-IOT-Stack,” 2015).
- *Complex Event Processing* (Perseo): utilizado para análise de dados de eventos, em tempo real, tornando possível responder de imediato a mudanças de condições, este modulo permite assim uma reação como por exemplo enviar e-mail ou uma mensagem (“Complex Event Processing (Perseo) - FIWARE-IOT-Stack,” 2015).
- *Identity Management* (IDM): o componente IDM é onde é apresentada a funcionalidade de login. É também onde as ações realizadas por qualquer utilizador são validadas e onde é feita a gestão de utilizadores (“Identity Management (IDM) - FIWARE-IOT-Stack,” 2015).

2.2.4. C2NET – IoT Hub

De acordo com (Ghimire, Melo, Ferreira, Agostinho, & Goncalves, 2015; Partners C2NET, 2015) o projeto C2NET procura a massificação de dispositivos IoT na industria, o que, por consequência resulta numa grande variedade de tecnologias, protocolos de comunicação e modelos de dados de resposta para a plataforma C2NET. Assim a plataforma C2NET pretende responder a este desafio através do C2NET IoT Hub (Figura 2.13), onde os protocolos IoT e o processamento de dados são suportados.

O *IoT Hub* é um *middleware* responsável por adquirir e despachar dados provenientes dos dispositivos físicos para a plataforma C2NET, permitindo assim que vários dispositivos estejam conectados em simultâneo, e tornando o *IoT Hub* uma central de observação para o mundo real. Esta solução conta na sua implementação com os mais utilizados protocolos de comunicação em dispositivos IoT.

Para um melhor entendimento deste *middleware*, os componentes que o constituem são brevemente explicados abaixo.

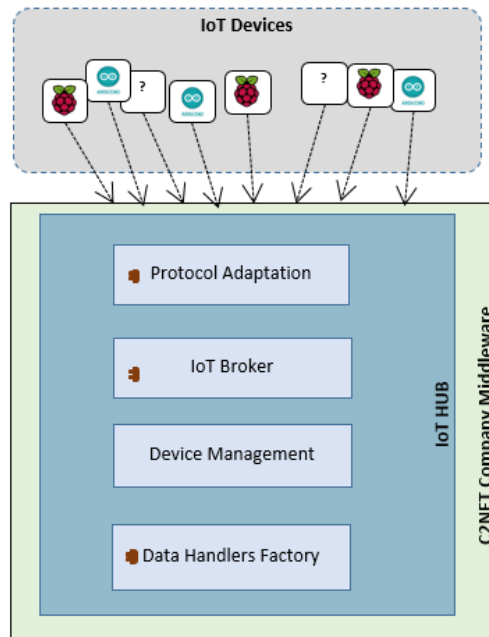


Figura 2.13 - C2NET - *IoT Hub* (Partners C2NET, 2015).

- *Protocol Adaptation and IoT Broker*: são responsáveis por providenciar a comunicação entre dispositivos IoT e os outros componentes do Hub. Estes componentes também permitem a conectividade de vários dispositivos com standards de comunicação diferentes, nomeadamente MQTT, XMPP, DDS e AMQP. Estes standards são traduzidos para o protocolo interno de comunicação do C2NET.
- *Device Management Component*: este componente é responsável pela integração de diferentes tipos de dispositivos com o Hub e também a gestão de todos os dispositivos ligados a este. O *Device Management* também providencia o estado do registo, conectividade e garantia de segurança de todos os dispositivos ligados à plataforma.
- *Data Handlers Factory*: este componente é responsável pelo pré-processamento de dados em tempo real, e também pela filtragem, fusão e agregação de dados provenientes de fontes.

2.2.5. Análise – Estratégias para a aquisição Automática de Dados

De forma a resumir e melhorar o processo de ponderação em relação a qual solução será a mais adequada a este trabalho, seguidamente é apresentada uma tabela comparativa (Tabela 2.3) das soluções enumeradas e descritas anteriormente.

Tabela 2.3 - Comparativo de Soluções IoT.

Nome	Tipo de Solução	Vantagens	Desvantagens
IoT-A	Arquitetura	Solução Geral e Abrangente	Dado ser uma arquitetura não existe implementação
OpenIoT	Plataforma	Permite desenvolvimento de aplicações direto na plataforma (<i>User Friendly</i>)	Muito Completa, Fechada
FIWARE IoT Stack	<i>Middleware</i>	Integra todos os Protocolos de comunicação com dispositivos IoT existentes	Muito Complexo
C2NET IoT HUB	<i>Middleware</i>	Solução leve e configurável	Não implementa todos os protocolos para dispositivos existentes

Numa visão geral das soluções apresentadas para a aquisição de dados observa-se que existem várias opções. Contudo estas soluções, embora variadas, pretendem resolver o mesmo problema levando à preocupação de qual será a mais adequada. O aspeto complexidade de implementação deve ser levado em consideração quando é necessário fazer uma escolha deste tipo.

No caso da arquitetura IoT-A, esta revela-se extremamente bem dividida e apresenta todas as preocupações necessárias para um sistema deste tipo. Dado o facto de ser uma arquitetura de referência, e por isso, altamente pormenorizada, traduz-se na inexistência de uma implementação exata desta solução.

Já no caso da solução *OpenIoT* esta revela-se uma solução igualmente bem dividida e completa. O *OpenIoT* oferece a possibilidade de desenvolvimento na própria plataforma, o que pode ser considerado um fator abonatório, contudo, e devido a esse facto esta torna-se uma solução extremamente fechada. Considerando que este trabalho pretende ser utilizado em cenários indústrias isso implicaria uma adaptação de todos os outros sistemas a esta plataforma.

A solução *FIWARE IoT Stack* apresenta-se como uma solução que permite ao programador, uma fácil integração de sensores e escalabilidade do sistema. Sendo estes pontos que tornam esta solução extremamente atrativa, a sua complexa arquitetura revela-se um fator negativo.

Por fim a solução *C2NET IoT Hub*, também se apresenta com os seus módulos e as funções dos mesmos bem definidas. Embora não tenha na sua implementação uma variedade de protocolos tão vasta como outras soluções estudadas, o facto de ser uma implementação leve que introduz

baixos consumos energéticos e facilmente configurável torna esta solução bastante apropriada ao trabalho que será desenvolvido.

Capítulo 3 – Acomodação por prioridades suportada por IoT

3.1. Caracterização do Problema por Prioridades

Neste capítulo é apresentada uma solução para melhorar o processo de armazenamento e transporte de mercadorias onde se identifica qual o tipo de solução para o problema de otimização de espaço presente tendo em conta as características do cenário. Para isso criou-se um sistema baseado em prioridades associadas a cada um dos itens, podendo estas ter ou não uma relação direta com as características dos itens a acomodar.

No contexto deste trabalho, a prioridade, é definida como o principal fator de diferenciação de um item face a todos os restantes. A prioridade do item tem um papel determinante na acomodação. O facto de um item ter uma prioridade superior a outros irá resultar na sua acomodação em detrimento de outros.

Alem de permitir a seleção de itens a acomodar, a prioridade do item também pode ser levada em consideração para determinar a ordem de entrada de um item no momento da acomodação, determinando qual o primeiro item a ser acomodado o que na ordem inversa também leva a que um item sem prioridade, seja acomodado no fim. A prioridade dos itens pode ser catalogada consoante várias características dos mesmos, tais como: custo, lucro, data de saída, data de entrega ou data de validade.

Embora exista a possibilidade de a prioridade ser diretamente proporcional ou estar diretamente relacionada com uma das características mencionadas anteriormente também existe a possibilidade de isso não acontecer. Por exemplo num caso em que se pretenda maximizar o lucro dos itens a acomodar, a seleção será feita de forma a que os itens a acomodar perfaçam o maior lucro possível. Contudo um item que ficaria excluído inicialmente da acomodação por ter um lucro menor do que os restantes, ou seja, uma menor prioridade, pode ser determinante para

um projeto ou cliente. Assim esta pode ser alterada, permitindo que o item integre o grupo de produtos selecionados para serem acomodados. Com a introdução da prioridade o utilizador terá um maior controlo do que é realmente importante numa determinada acomodação.

Com base no conceito apresentado, foi criado neste trabalho uma árvore de suporte ao processo de decisão. Apresentada na Figura 3.1 e composta por três níveis. No primeiro é avaliada a prioridade, no segundo a heterogeneidade dos itens, ou seja, a diversidade de tamanhos e formas destes, e no terceiro é identificado o problema correspondente:

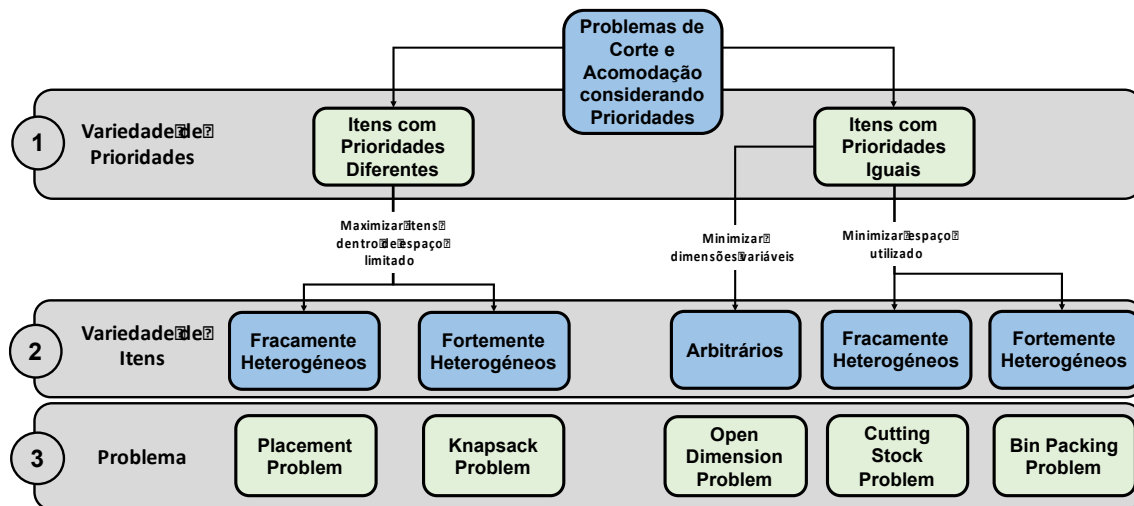


Figura 3.1 – Árvore de classificação baseada em prioridades

Nesta metodologia o processo de classificação começa, nível 1, por avaliar os níveis de prioridade existentes entre todos os itens com o intuito de identificar a existência ou não de diferentes níveis de prioridade. Desta forma é possível conhecer se é necessário efetuar ou não uma seleção de itens, sendo assim feita a opção entre os dois ramos principais.

Num cenário em que existam diferentes prioridades, estas serão levadas em consideração. No caso de nem todos os itens poderem ser acomodados por falta de espaço, a principal preocupação será a maximização de itens prioritários dentro do espaço disponível.

Já no caso de todos os itens terem a mesma prioridade não será possível essa seleção, pois dado que tem a mesma prioridade é implícito que todos terão de ser acomodados passando-se para o segundo nível de seriação, onde a principal preocupação é a minimização do espaço necessário para a acomodação de todos os itens.

Seguidamente no ramo dedicado aos problemas em que as prioridades são diferentes, os itens são classificados quanto à sua heterogeneidade, nível 2, de forma a que seja possível identificar a variedade de formas existentes de entre os itens a acomodar. Assim os resultados possíveis poderão ser que os itens a acomodar são fracamente heterogéneos o que se traduz num problema denominado **Placement Problem**, ou são fortemente heterogéneos o que resulta no problema **Knapsack Problem**, nível 3.

No ramo onde as prioridades são iguais para todos os itens e tendo em conta o objetivo de minimizar o espaço necessário para a acomodação de todos os itens, no caso de a variedade dos itens ser arbitrária, nível 2, e o espaço para ter a capacidade de variar as suas dimensões físicas é identificado o problema denominado **Open Dimension Problem**, nível 3. Já no caso de o espaço não ter essa capacidade é avaliada a heterogeneidade dos itens, que à semelhança do caso anterior, podem ser fracamente ou fortemente heterogéneos, nível 2, resultando assim nas classificações **Cutting Stock Problem** e **Bin Packing Problem** respetivamente, nível 3.

A tabela apresentada em seguida resume a classificação apresentada (Tabela 3.1).

Tabela 3.1 – Tabela resumo da classificação de Problemas.

Problema	Prioridades	Heterogeneidade	Missão/Tarefa
Placement problem	Diferentes prioridades	Fraca heterogeneidade	Maximizar o numero de itens prioritários num espaço limitado
Knapsack Problem	Diferentes prioridades	Forte heterogeneidade	Maximizar o numero de itens prioritários num espaço limitado
Open Dimension Problem	Prioridades iguais	Heterogeneidade Arbitraria	Minimizar as dimensões do espaço ocupado
Cutting Stock Problem	Prioridades iguais	Fraca heterogeneidade	Minimizar o espaço necessários
Bin Packing Problem	Prioridades iguais	Forte heterogeneidade	Minimizar o espaço necessários

3.2. Sistema de Otimização Inteligente

Tendo como base a metodologia para a classificação de problemas apresentada anteriormente, foi criado um Sistema de Otimização Inteligente, que dado um determinado cenário, entenda-se itens a acomodar e prioridades associadas aos mesmos, determina qual a forma mais eficiente de acomodar esses itens. De forma a conceptualizar uma arquitetura para o sistema proposto, foi utilizado como base as soluções IoT estudadas no capítulo anterior, e será descrita em seguida.

A arquitetura do sistema apresentada na Figura 3.2 é dividida em dois níveis, Aplicacional e Físico, com os quais é identificável os módulos do sistema responsáveis pela aquisição de dados, Nível Físico, e os módulos responsáveis pela otimização, Nível Aplicacional. Assim num todo a arquitetura apresenta seis módulos: Interface de Utilizador, Classificador de Cenários, Optimizador de Espaço, Base de Dados, Mediador IoT e Rede de Sensores. De notar que na imagem apresentada o conjunto Classificador de Cenários e o Optimizador de Espaços, é denominado Centro de Decisão, isto deve-se ao facto de estes dois módulos serem responsáveis por todo o processamento da informação necessário à otimização.

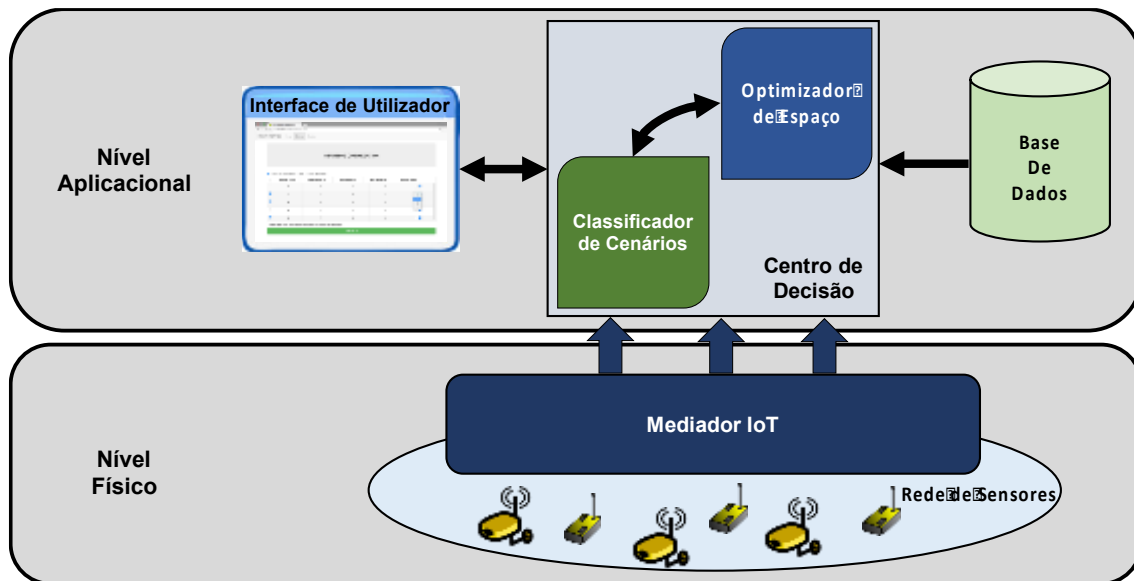


Figura 3.2 – Arquitetura do Sistema de Otimização Inteligente.

Seguidamente cada um dos módulos constituintes da arquitetura/sistema é apresentado individualmente para clarificação das funcionalidades dos mesmos no sistema.

3.2.1. Interface de Utilizador

A Interface de Utilizador, IU, permite ao utilizador visualizar quais os itens que estão disponíveis para acomodação e recolher quais os itens que o utilizador pretende acomodar. A IU também tem como responsabilidade permitir que o utilizador identifique qual a prioridade que pretende que seja atribuída a cada um dos itens a acomodar. Além disto é responsável por questionar o utilizador de como proceder no caso de não ter a informação necessária para resolver o problema por si só.

3.2.2. Classificador de Cenários

O módulo Classificador desempenha a função de decisão e/ou classificação do sistema. Este módulo é responsável pelo processamento da informação de forma a possibilitar a decisão de qual o problema que corresponde às características do cenário. Nesta solução essa classificação é feita através da utilização da árvore de suporte à decisão mencionada anteriormente. Assim este módulo é responsável por classificar o tipo de problema de acomodação apresentado com base nos itens e prioridades seleccionadas pelo utilizador na IU.

3.2.3. Optimizador de Espaço

O módulo Optimizador desempenha a função de cálculo da posição dos itens. Através da classificação feita anteriormente pelo Classificador, este módulo determina qual a posição que cada item deve ocupar dentro do espaço de acomodação através do algoritmo correspondente

à classificação. Contudo, para tal o Optimizador necessita também de informação referente às dimensões e prioridade de cada item bem como das dimensões do espaço onde será feita a acomodação. Essas informações são fornecidas pela Classificador no caso dos itens e pela Base de Dados no caso do espaço.

3.2.4. Base de Dados

Este módulo é responsável por armazenar a informação referente aos itens a acomodar, bem como a informação referente ao espaço de acomodação. A Base de Dados, BD, também é responsável por guardar o histórico das arrumações anteriores. Este modulo será responsável por providenciar as informações referentes aos itens ao modulo IU e as informações referentes ao espaço ao Optimizador.

3.2.5. Mediador IoT

O modulo Mediador é o modulo responsável por permitir a ligação entre o nível físico e o nível aplicacional no sistema. Aqui são tratados os problemas como a transferência de dados e a fiabilidade dos mesmos. Para tal os sensores ou dispositivos devem ser conectados ao Mediador permitindo assim que este encaminhe os dados para o Sistema.

3.2.6. Rede de Sensores

A rede de sensores é o modulo responsável pela aquisição dos dados necessários à arquitetura/sistema de otimização no mundo real. Os sensores que constituem esta rede permitem a aquisição das dimensões dos itens de forma a permitir uma caracterização destes a três dimensões.

3.2.7. Funcionalidade Integrada

De forma a demonstrar o funcionamento da arquitetura/sistema, e, as operações entre os módulos, em seguida são apresentadas na Figura 3.3 as trocas de informação constituintes de num processo de otimização.

Como pode ser observado seguindo a numeração associada às trocas de informação, o processo inicia com a aquisição de informação por parte do módulo Rede de Sensores. Essa informação é encaminhada para o Mediador IoT, dado que este é o modulo responsável por fazer a comunicação entre os níveis Físico e Aplicacional.

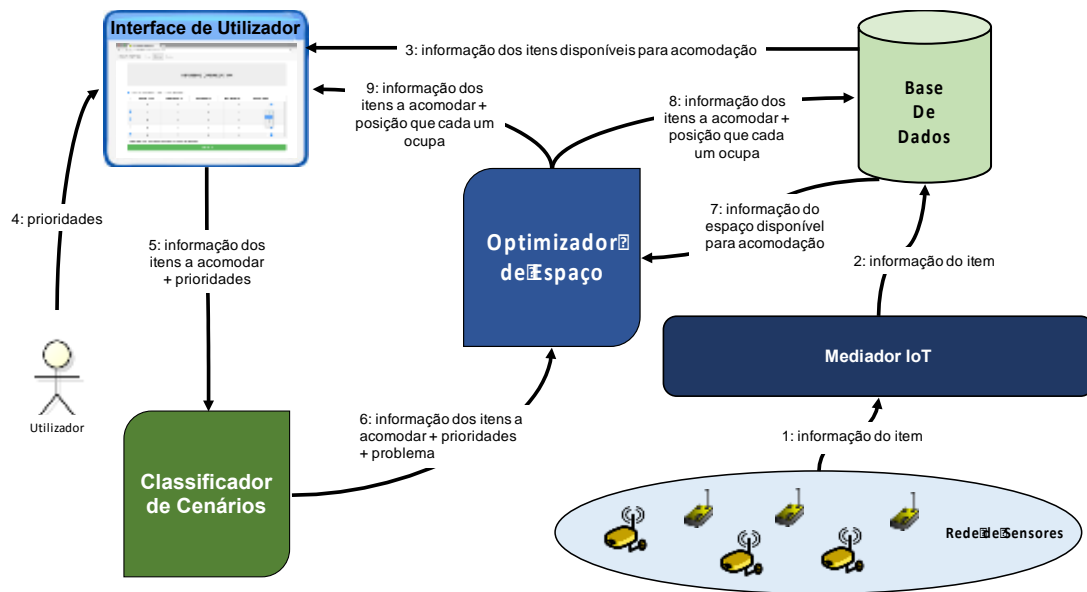


Figura 3.3 – Interação de módulos no processo de otimização.

Assim com a informação no Nível Aplicacional, entenda-se Base de Dados populada, a informação é requisitada pelo módulo Interface de Utilizador de forma a que nesse módulo o utilizador possa associar as prioridades aos itens que pretende acomodar. Após isso a informação do grupo de itens a acomodar é enviada para o Classificador onde dada a variedade de itens e de prioridades o cenário é classificado.

Por fim de forma a possibilitar o cálculo da posição de cada item, o Optimizador recebe do Classificador a informação relevante para o cálculo, ou seja, a classificação obtida, e a informação dos itens (dimensões e prioridade) que devem ser acomodados. O Optimizador irá também receber para efetuar o cálculo das posições a informação do espaço para a acomodação, que como mencionado anteriormente é proveniente da Base de Dados.

Após a finalização do cálculo as posições são enviadas para o Base de Dados (Tabela de Histórico) e as mesmas posições são partilhadas com o utilizador através da Interface de Utilizador de forma a informar como proceder para concretizar a otimização.

3.3. Prova de Conceito - Implementação

Utilizando como base a Classificação e Arquitetura introduzidas acima, neste ponto é apresentada a implementação concebida para comprovar a veracidade da hipótese desta dissertação. Será feita uma breve descrição das funcionalidades implementadas em cada um dos módulos bem como as tecnologias utilizadas nos mesmos.

3.3.1. Interface de Utilizador:

Este modulo foi implementado usando como base a linguagem HTML¹ de forma a possibilitar a sua consulta online, e, com recurso à biblioteca Bootstrap² para que essa consulta online possa ser efetuada através de qualquer dispositivo móvel. Assim quando o utilizador acede à solução é-lhe apresentada a Interface de Utilizador que tem o aspeto representado na Figura 3.4.

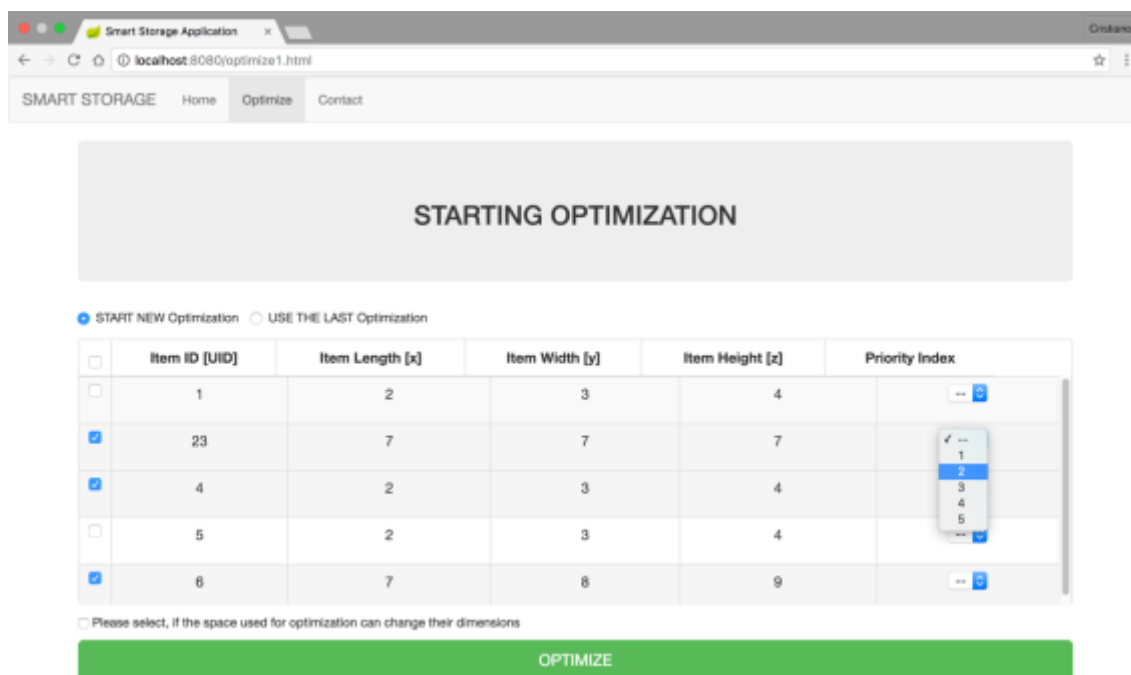


Figura 3.4 - Interface de Utilizador³

Com recurso a *radio buttons*, foi implementada a funcionalidade que permite ao utilizador escolher entre a execução de uma nova otimização ou utilizar os dados já selecionados na otimização anterior e a esta acrescentar novos dados. Assim permite-se ao utilizador dar continuidade a uma otimização já existente sem que seja necessário refaze-la.

Posteriormente o utilizador tem acesso à tabela de itens (Figura 3.). Nesta tabela é possível visualizar todos os itens disponíveis para otimização/acomodação e fazer a seleção destes itens através da utilização das *checkboxes* localizadas na coluna à esquerda.

¹ Linguagem de programação baseada em marcações, em inglês *tags*, utilizada na descrição de documentos/páginas web, (http://www.w3schools.com/html/html_intro.asp), (The HTML SourceBook, Ian S. Graham, 1995).

² Biblioteca para desenvolvimento responsivo de projetos web para dispositivos móveis (<https://getbootstrap.com/>).

³ A implementação da interface foi realizada na língua inglesa de forma a permitir que este sistema, não apresente qualquer entrave à sua implementação na industria internacional.

<input type="checkbox"/>	Item ID [UID]	Item Length [x]	Item Width [y]	Item Height [z]	Priority Index
<input checked="" type="checkbox"/>	1	2	3	4	--
<input checked="" type="checkbox"/>	23	7	7	7	--
<input type="checkbox"/>	4	2	3	4	--
<input checked="" type="checkbox"/>	5	2	3	4	--
<input type="checkbox"/>	6	7	8	9	--

Figura 3.5 - Tabela de Itens.

A tabela de itens apresenta também na coluna mais à direita a coluna responsável pela aquisição das prioridades associadas (Figura 3.), onde o utilizador terá de atribuir manualmente o índice de prioridade que deseja dar a cada item.

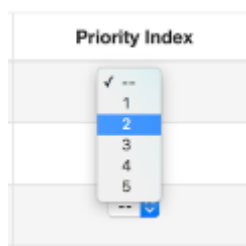


Figura 3.6 - Coluna de índice de prioridade.

Na interface foi também implementado com recurso a uma *checkbox* uma forma do utilizador introduzir a informação se o espaço que vai ser utilizado para a acomodação permite a variação das suas dimensões. Assim caso o espaço tenha essa propriedade o utilizador deve seleccionar esta opção.

3.3.2. Classificador de Cenários:

O módulo Classificador, como mencionado anteriormente é responsável pela classificação das informações fornecidas pelo utilizador através da IU, e com isso determinar qual o tipo de problema correspondente.

Assim, para alcançar esse resultado, foi implementado através da utilização da linguagem JavaScript⁴ um classificador baseado na árvore de suporte à classificação apresentada anteriormente. Essa implementação, bem como a origem dos dados para esta, é apresentada através do diagrama de fluxo (Figura 3.) seguinte.

⁴ Linguagem de programação baseada em scripts, os quais são processados no pela maquina do utilizador, por exemplo, no *web browser*, (JavaScript: The Definitive Guide, David Flanagan, 2006).

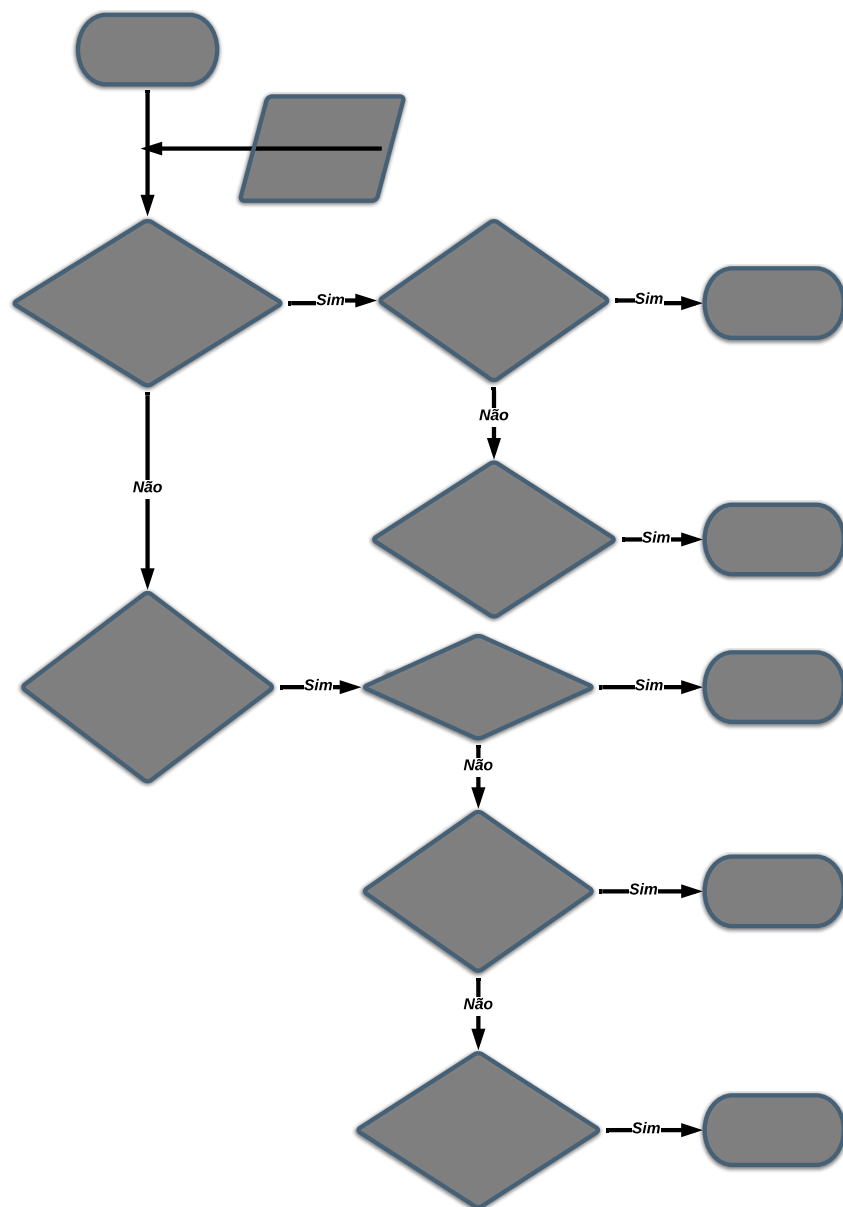


Figura 3.7 - Fluxograma da decisão no Classificador

De forma a possibilitar esta classificação foi implementado um índice de prioridades que tem valores entre 1 e 5 sendo o **item com prioridade 1 menos importante que o item de prioridade 5**, e estas recebidas pelo Classificador num vetor com o par (identificador único, prioridade), enviado pelo modulo Interface de Utilizador. No caso de o utilizador não seleccionar qualquer prioridade estas serão tratadas como prioridades iguais.

O módulo Classificador é capaz de proceder à classificação através da análise das prioridades, que consiste em verificar se as prioridades presentes são todas iguais ou não. Contudo e para finalizar a análise, a heterogeneidade dos itens a acomodar é avaliada comparando coordenada a coordenada de todos os itens seleccionados, formando assim classes de heterogeneidade.

Por fim a informação a ser partilhada com o Optimizador é organizada em dois vetores. O primeiro (Figura 3.) conta em cada posição com a caracterização da classe representada pelas medidas x, y e z e um outro vetor onde cada posição deste contem o identificador único (UID) de um item pertencente a essa classe, garantindo assim a divisão por classes, ou seja, uma divisão dos itens segundo o seu tamanho e/ou forma. O segundo vetor, vetor de prioridades (Figura 3.) apresenta uma organização similar ao anterior, sendo que cada posição desse vetor não corresponde a uma classe mas sim a uma prioridade, ou seja cada posição contem o nível de prioridade e um vetor com os UIDs dos itens caracterizados com essa mesma prioridade pelo utilizador. Contudo a primeira posição do vetor de prioridades é a posição onde é introduzido o problema de otimização a ser executado pelo Optimizador.

Classe 1 X = ... Y = ... Z = ...	Classe 2 X = ... Y = ... Z =
UID1
UID2
UID3	...	
...		

Figura 3.8 - Vetor de Classes

Problema	Prioridade 1 Valor = ...	Prioridade 2 Valor =
	UID1
	UID2
	UID3	...	
	...		

Figura 3.9 - Vetor de prioridades

3.3.3. Optimizador de Espaço

O módulo Optimizador ponderado para esta implementação, recebe os vetores de classes e prioridades já descritos bem como o um outro no qual esta contida a informação dos espaços disponíveis para a acomodação dos itens (Figura 3.). A informação contida em cada posição desse vetor é as suas dimensões (x, y e z) e a sua quantidade disponível, de forma que no caso de existir mais que um espaço igual não será necessário criar uma nova posição no vetor.

Espaço 1 X = ... Y = ... Z = ... Quant. = ...	Espaço 2 X = ... Y = ... Z = ... Quant. =
--	--	-----

Figura 3.10 - Vetor de Espaços disponíveis.

Com estas informações, o Optimizador é capaz de efetuar o cálculo da posição que cada item deve ocupar dentro espaço. De forma a clarificar mais a função que este modulo desempenha é apresentado um pseudo-código (Figura 3.) onde se exemplifica o método de funcionamento do Optimizador para a calculo das posições num problema *Bin Packing Problem* com recurso ao algoritmo *First Fit Decreasing* (Yesodha & Amudha, 2012).

É considerado que todos os itens têm a mesma prioridade dado tratar-se de um *Bin Packing Problem*. Considera-se ainda a existência de três classes, cada uma delas com um item, e três espaços com dimensões diferentes.

```

Vetores de entrada: vetor Classes, vetor Prioridades, vetor Espaços
Variáveis de entrada: posição Classe, posição Prioridade, posição Espaço = 0

INICIO
  Ler vetor Prioridades na posição Prioridade
  Se posição Prioridade = Bin Packing Problem
    Para Posição Espaço no vetor Espaços até Ultima Posição Espaço do vetor Espaços
      Ler vetor Classes na posição Classe
      Ler vetor Espaços na posição Espaço
      Se Item na posição Classe do vetor Classes ≤ Espaço na posição Espaço do vetor Espaços
        Subtrair Volume de Item ao Volume do Espaço
        Guardar resultado em posição Espaço
        Incrementar posição Classe
        posição Espaço = 0
      Fim Se
      Se Não
        Incrementar posição Espaço
      Fim Se Não
    Fim Para
  Fim Se
FIM

```

Figura 3.11 - Pseudo-código do Optimizador para o problema *Bin Packing Problem*.

O funcionamento do Algoritmo apresentado é extremamente simples, pois neste caso, como o problema Bin Packing Problem considera que todos os itens são acomodados, o espaço é sempre suficiente e o algoritmo resume-se à busca de um espaço grande o suficiente para a acomodação do item. Caso o espaço disponível não seja suficiente passa para o espaço seguinte.

Assim, de forma a possibilitar a validação deste sistema o módulo Optimizador implementado permite o calculo da otimização em Cenários correspondentes ao *Bin Packing Problem*, com recurso ao algoritmo apresentado. Contudo, o módulo não conta com os algoritmos para o calculo dos restantes cenários, devido ao facto de estes necessitarem de uma otimização à questão das prioridades, trabalho esse que será desenvolvido posteriormente.

3.3.4. Base de Dados

O módulo Base de Dados foi implementado com recurso a MySQL⁵. Assim para a administração da base de dados foi usada a ferramenta phpMyAdmin⁶. Na figura abaixo (Figura 3.) é apresentada a tabela com as dimensões dos itens.

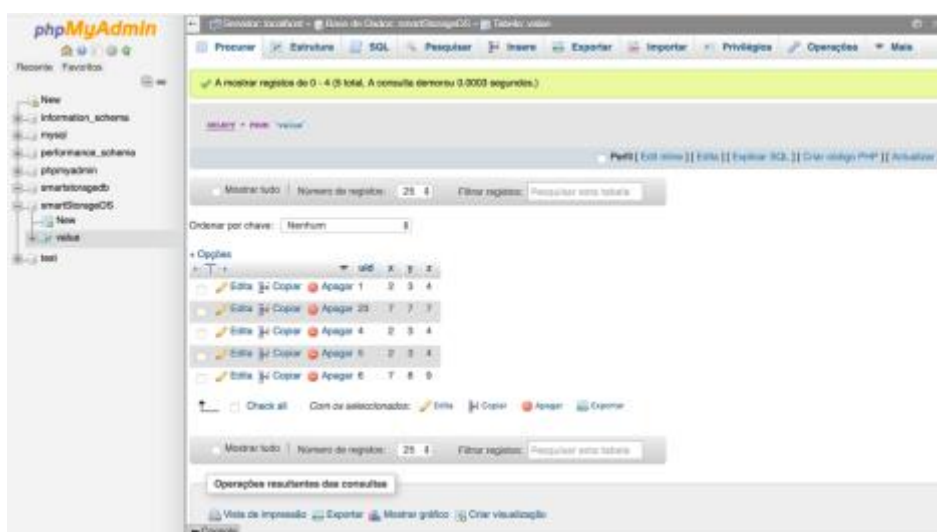


Figura 3.12 - Tabela de Itens.

A tabela de itens implementada na base de dados é composta por 4 colunas e estas comportam os valores referentes ao UID, que é único para cada item, medida do item segundo a coordenada X (x), medida do item segundo a coordenada Y (y) e medida do item segundo a coordenada Z (z). Contudo, a Base de Dados necessita de ser populada, sendo que tal acontece através da rede de sensores ou dispositivos associados ao sistema.

À imagem da tabela de itens apresentada, foi também implementada uma tabela para os espaços, a qual conta com as mesmas três colunas para as dimensões x, y e z, mas não tem qualquer identificador. No caso da tabela de espaços foi introduzida uma coluna para as quantidades. Essa coluna pretende manter atualizadas as quantidades existentes de cada espaço. Este detalhe foi levado em consideração pois dado que os espaços podem representar

⁵Banco de Dados de código aberto, (<https://www.oracle.com/br/products/mysql/overview/index.html>).

⁶ Ferramenta para a administração de Bancos de dados MySQL, (<https://www.phpmyadmin.net/>).

camiões, seria interessante ter detalhado nessa tabela a frota de uma empresa de logística podendo ajudar no controlo de quantos estariam em trânsito e quantos estariam no cais.

Por fim foi criada uma tabela de histórico, onde são guardadas as otimizações efetuadas. Esta tabela tem na sua primeira coluna a informação de qual o problema utilizado para o cenário e nas restantes os itens que compunham esse cenário. Nesta tabela cada linha representa uma otimização.

3.3.5. Mediador IoT

O Mediador IoT utilizado nesta implementação foi a solução IoT estudada nesta dissertação, IoT HUB, que faz parte do projeto C2NET e para o qual este trabalho contribui. Como indicado anteriormente este modulo é responsável por estabelecer a ligação entre o nível Físico e Aplicacional do sistema.

Assim, para corresponder às necessidades desta implementação este modulo foi ligado via porta Serie à rede de sensores, e configurado para utilizar o protocolo de comunicação Serie com recurso às bibliotecas *gnu.io*⁷, de forma a escutar a porta definida e com isto garantindo a ordem dos dados.

O IoT Hub também foi configurado tendo em conta as características dos sensores e as suas leituras, ou seja, de forma a identificar a informação pretendida que no caso deste trabalho é o UID do item e as suas três dimensões em centímetros nas coordenadas x, y e z. Esta informação é adquirida pelo Mediador através de um serviço REST.

Outra configuração que foi introduzida para a perfeita adaptação ao pretendido neste trabalho foi a configuração do *rowmapper* que é responsável por fazer a ligação entre o modelo do sensor e as colunas respetivas na Base de Dados. De salientar que estes dados são transmitidos através da utilização da tecnologia JSON.

3.3.6. Rede de Sensores

Antes de dar início à descrição dos constituintes e das funcionalidades que constituem o modulo Rede de Sensores, é fulcral clarificar que o Hardware mencionado foi o adotado para esta implementação, contudo outro Hardware pode ser igualmente utilizado desde que garantido que as funcionalidades abaixo descritas são suportadas pelo mesmo.

A rede de sensores utilizada para constituir o sistema foi desenvolvida usando como base o microcontrolador Arduino Mega⁸ (Figura 3.-a). A escolha deste microcontrolador deve-se à maior

⁷ Bibliotecas para a comunicação com a rede de sensores, (<http://playground.arduino.cc/Interfacing/Java>).

⁸ <https://www.arduino.cc/en/Main/arduinoBoardMega>

oferta de número de portas de *input*, em português entrada, bem como as suas velocidades de operação superiores.

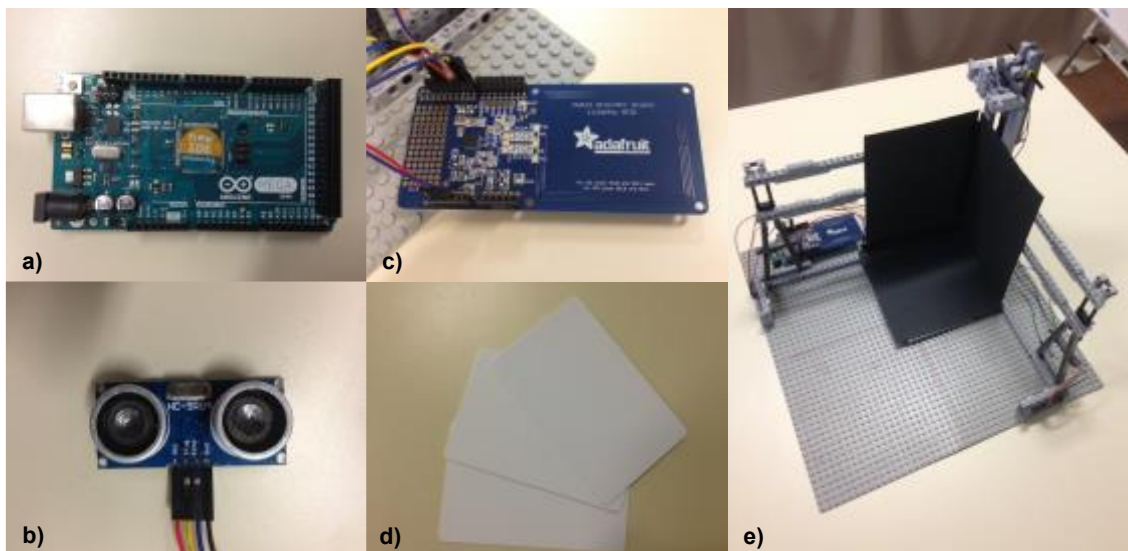


Figura 3.13 – a) Arduino Mega, b) Sensor ultrassónico HC-SR04, c) Shield NFC/RFID NP532, d) Mifare Classic Card e e) Rede de Sensores Final.

De forma a serem feitas leituras das dimensões segundo as coordenadas x, y e z mencionadas anteriormente, foram conectados ao controlador três sensores ultrassónicos de modelo HC-SR04(Figura 3.-b). Estes sensores foram os escolhidos devido ao seu baixo custo, à sua facilidade de integração com o controlador e devido a permitirem leituras entre 2 centímetros e 4 metros com uma precisão de 3 milímetros, o que se revelou serem características suficientes dado o objetivo da implementação.

A rede de sensores contou também com a introdução de uma Shield NFC/RFID⁹ (Figura 3.-c) que através do funcionamento com cartões Mifare Classic card¹⁰ (Figura 3.-d) permite ao guardar a informação, entenda-se medidas nas coordenadas x, y e z, nesses mesmos cartões e obter o UID proveniente deste, levando a que este represente um cartão de identificação do item e que o dispositivo seja utilizado uma única vez em cada item, pois uma vez lido pela rede de sensores este passa a constar no sistema e a ter uma identificação (cartão) que permite ter acesso às suas características.

Por fim de forma a que os sensores estejam bem colocados para uma leitura precisa, estes foram fixados a uma estrutura e direcionados ao ponto de origem, resultando assim na rede de sensores pretendida (Figura 3.-e). De referir que esta rede de sensores é uma versão experimental tanto em tamanho como em funcionalidades do que seria uma rede de sensores preparada para integração em cenários industriais reais.

⁹ Adafruit Shield NFC PN532, <https://www.adafruit.com/product/789>

¹⁰ http://www.nxp.com/products/identification-and-security/mifare-ics/mifare-classic:MC_41863

Capítulo 4 - Testes e Validação

4.1. Metodologia de Teste

Encontrar erros na implementação do sistema através da utilização da experimentação é também conhecida como um processo de teste. Este processo pode ser realizado num ambiente especial onde o uso normal e excepcional do sistema pode ser simulado. Embora os testes mostrem ao responsável pela conceção do sistema a presença de erros e não a sua ausência, ele não garante a correção completa de uma implementação (Tretmans, 2001).

Para cada uma das áreas de aplicação existem diferentes métodos para testar a adaptabilidade das soluções aos problemas (Onofre, 2007), através da utilização do standard internacional para o teste de compatibilidade para sistemas abertos, ou seja, o ISSO-9646 ("ISO/IEC 9646," n.d.): "OSI Conformance Testing Methodology and Framework".

Uma aproximação deste standard é demonstrada na figura abaixo (Figura 4.1). Para testar a hipótese um conjunto de testes devem ser definidos. Contudo para esta ser testada esta deve ser implementada como prova de conceito. Essa implementação não necessita ser completamente funcional, mas deve preencher todos os requisitos na arquitetura proposta. Os resultados dos testes efetuados são analisados, levando assim a um veredicto da conformidade do sistema em teste para com os requisitos definidos inicialmente (Tretmans, 2001).

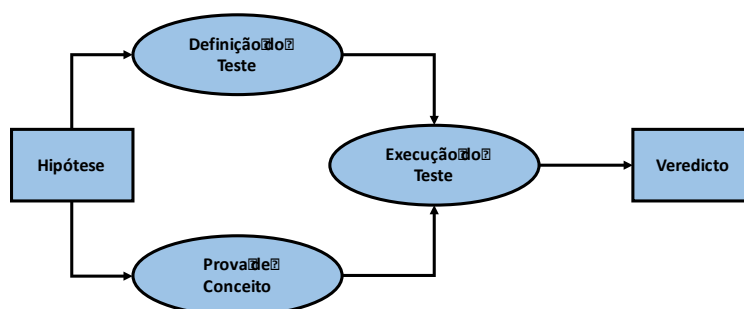


Figura 4.1 - Processo de Teste

Para garantir que os testes podem ser executados, estes são especificados tendo em conta uma bem conhecida notação, que é independente de qual quer implementação e aceite globalmente. Assim o standard TTCN-2, onde cada ambiente de teste é definido por uma sequencia de eventos que ocorrem durante o teste (Tretmans, 1992).

Este é definido em forma de tabela, e aumenta a identificação de eventos sendo estes indicados numa cadeia sucessiva. Os eventos alternativos são definidos usando a mesma identificação. O ponto de exclamação usado antes de um evento indica que é uma ação e o ponto de interrogação indica que é uma condição. A sequência termina com a especificação do veredicto que é atribuído quando a execução da sequência termina que pode ser apresentado como: "Sucesso", "Falha" ou "Inconclusivo". Cada tabela de teste possui um cabeçalho, onde é definido o nome do teste, o seu objetivo, as entradas necessárias durante a execução do teste, e as saídas resultantes.

Um exemplo é demonstrado na Tabela 4.1. Esta exemplifica a passagem de informação entre o Dispositivo responsável pela aquisição de dados e o Mediador que coloca essa informação no sistema. Em primeiro lugar é estabelecida a comunicação entre o Dispositivo e o Mediador, e verifica se existe transmissão de informação, se não existir informação no Mediador então o veredicto será automaticamente "INCONCLUSIVO" pois este facto significa que a ligação não foi conseguida não podendo ser testada a transmissão de dados. Por outro lado, caso exista informação esta deve ser analisada e no caso desta corresponder à informação esperada o veredicto é "SUCESSO", por fim no caso dessa informação não corresponder à informação esperada, o veredicto é "FALHA".

Tabela 4.1 - Exemplo de Tabela de teste baseada em TTCN

Transmissão de informação entre o Dispositivo e o Mediador		
Nome do Teste:	Teste à transmissão de informação entre o Dispositivo e o Mediador	
Proposta:	Verificar se o dispositivo e o Mediador estão em comunicação e se os valores lidos no dispositivo chegam ao Mediador	
Entradas:	E1: [uid,x,y,z] enviado, E2: [uid,x,y,z] esperado	
Saídas:	S1: [uid,x,y,z] resultado	
Numero de Linha	Ambiente	Veredicto
1	! Passagem de Informação do Dispositivo para o Mediador	
2	?..Apresentação da Informação esperada (S1), logo existe conexão	
3	? Resultado (S1) igual ao esperado (E1)	SUCESSO
4	? Resultado (S1) diferente do esperado (E1)	FALHA
5	? Não há passagem de informação do Dispositivo para o Mediador	INCONCLUSIVO

De forma a apresentar os resultados dos testes executados, outra tabela também usada é a demonstrada na Tabela 4.2 que tal como a tabela anterior reflete o exemplo anterior. Nesta

segunda tabela cada linha representa um caso de teste específico e o número de veredictos resultantes da tabela TTCN, para os resultados esperados e obtidos.

Tabela 4.2 - Tabela exemplo caso de teste

Teste	Entrada		Saída	Resultado (Número de Linha)	
	E1: [uid,x,y,z] enviado	E2: [uid,x,y,z] esperado	S1: [uid,x,y,z] recebido	Esperado	Obtido
1	[4;2;3;4]	[4;2;3;4]	[4;2;3;4]	(3)	(3)
2	[23;7;7;7]	[23;7;7;7]	[23;7;7;7]	(3)	(3)
3	Sem dados	Sem dados	Sem dados	(5)	(5)

4.2. Definição e Execução de Testes

Acompanhando o método mencionado acima os casos de teste vão ser divididos segundo duas áreas, sendo estas:

- Aquisição de dados: Neste caso, é estudada a aquisição de dados e se a sua transmissão para o Sistema de Otimização Inteligente. Começando por ser testado se a rede de sensores efetua leituras corretas, sendo seguido do teste à transmissão de dados entre o dispositivo e o Mediador e por fim o teste à transmissão desses dados entre o Mediador e o Sistema (Base de Dados).
- Otimização de espaço: Neste caso, é estudado se o sistema identifica o problema que melhor se adapta e o resolve. Para isso será testado a consulta da Base de Dados por parte da Interface, seguindo-se do teste à aquisição de informação na interface. Em terceiro lugar será testado o processo de decisão do problema e por fim é testado o cálculo das posições que devem ser ocupadas por cada item e a apresentação dessas posições ao utilizador.

4.2.1. Definição do Teste: Aquisição de dados

Este teste é efetuado de forma a garantir que o sistema implementado possui efetivamente a capacidade para adquirir dados referentes aos itens e transmitir esses mesmos dados ao sistema. É testado se os dados são capturados corretamente, e, se estes são entregues da mesma forma ao sistema. Neste teste fica implícito o facto de a ligação entre a rede de sensores e o Mediador e depois entre o Mediador e o sistema ser conseguida de forma eficiente.

Para a execução do teste as entradas consideradas foram, a informação colecionada, que é constituída pelas medidas (x, y e z), e o UID do item, sendo que esta informação será apresentada segundo a forma [UID, x, y, z]. Este teste tem como saída as informações

introduzidas na entrada pois estas devem ser entregues ao nível aplicacional que será testado em seguida.

Este teste apresentará falha no seu veredicto em dois casos, no caso de não serem entregues quaisquer dados após ser efetuada uma leitura por parte da rede de sensores ou no caso de os dados entregues ao sistema serem diferentes dos adquiridos. A tabela TTCN referente a este teste é apresentada abaixo na Tabela 4.3.

Tabela 4.3 – Definição do Teste: Aquisição de Dados.

Aquisição de Dados		
Nome do Teste:	Teste à Aquisição de dados	
Proposta:	Verificar se a informação adquirida pelo corretamente e se esta é corretamente entregue ao sistema	
Entradas:	E1: [uid,x,y,z] lido, E2: [uid,x,y,z] esperado	
Saídas:	S1: [uid,x,y,z] entregue	
Numero de Linha	Ambiente	Veredicto
1	! Leitura no Dispositivo da informação(E1).	
2	? Apresenta [uid,x,y,z] lido ao sistema (S1) logo a ligação conseguida.	
3	! Comparar [uid,x,y,z] entregue (S1) com [uid,x,y,z] esperado (E3).	
4	? [uid,x,y,z] entregue (S1) igual ao [uid,x,y,z] esperado (E2)	SUCESSO
5	? [uid,x,y,z] entregue (S1) diferente do [uid,x,y,z] esperado (E2)	FALHA
6	? Não apresenta [uid,x,y,z] lido ao sistema	FALHA

4.2.2. Execução do Teste

Este teste foi reproduzido mais do que uma vez com itens diferentes de forma a permitir a aquisição de dados diferentes e com isso promover a consistência dos resultados. Todos os dados foram registados na Tabela 4.4 apresentada abaixo.

Para cada um dos itens disponíveis e submetidos a este teste é esperado que as suas leituras correspondam aos valores apresentados na tabela.

Tabela 4.4 - Execução do Teste: Aquisição de Dados

Teste	Entrada		Saída	Resultado (Numero de Linha)	
	E1: [uid,x,y,z] lido	E2: [uid,x,y,z] esperado	S1: [uid,x,y,z] entregue	Esperado	Obtido
1	[4;2;3;4]	[4;2;3;4]	[4;2;3;4]	(4)	(4)
2	[23;7;7;7]	[23;7;7;7]	[23;7;7;7]	(4)	(4)

3	Sem Item	Sem Item	Sem Dados	(6)	(6)
---	----------	----------	-----------	-----	-----

Após a execução dos testes, o veredicto obtido foi de SUCESSO em todos os testes com exceção do teste no qual não existia item, e que por isso não houve informação entregue. Esse facto já era esperado pois não havendo presença de um item a sua informação não pode ser lida e conseqüentemente não pode ser entregue. Já no teste apresentado na linha 1, pode ser comprovado o veredicto obtido é de sucesso, pois a informação do item adquirida, [4, 2, 3, 4], através da rede de sensores é exatamente a informação entregue ao nível aplicacional tal como esperado. Observando-se um veredicto de SUCESSO em todos os testes efetuados.

4.2.3. Definição do Teste: Otimização de Espaço

Este teste é executado de forma a garantir que o nível aplicacional apresentado permite que seja determinado o problema correspondente ao cenário apresentado bem como o calculo da posição final de cada item no espaço reservado à acomodação. Será testado se o Classificador executa uma correta classificação bem como se o calculo das posições por parte do Optimizador é executado corretamente.

Para a execução deste teste as entradas necessárias são a informação dos itens a acomodar, representada por [Item1,Item2,Item3], o resultado esperado para o cenário à saída do Classificador e por fim o resultado esperado para as posições finais a ocupar por cada item, onde, será representado cada espaço utilizado para acomodação por parênteses retos e sendo colocado no seu interior os itens que foram acomodados nesse espaço. Tendo como saídas o resultado da classificação e o resultado das posições dos itens.

Este teste apresentara FALHA no seu veredicto em três situações, no caso de não haver classificação possível, no caso da classificação apresentada ser diferente da esperada e no caso do calculo de as posições ser diferente do esperado. À semelhança do teste anterior a tabela TTCN correspondente é apresentada na Tabela 4.5.

Tabela 4.5 - Definição do Teste: Otimização de espaço

Otimização de espaço		
Nome do Teste:	Teste à Otimização de espaço	
Proposta:	Verificar se a classificação e calculo das posições dos itens é correta.	
Entradas:	E1: Informação dos itens a acomodar, E2: Classificação Esperada, E3: Posições dos itens esperado	
Saídas:	S1: Classificação, S2: Posições dos itens	
Numero de Linha	Ambiente	Veredicto
1	! Informação presente na Interface(E1).	
2	? Apresenta classificação e posições para os itens (S1 e S2).	

3	! Comparar classificação (S1) com Classificação (E2).	
4	? Resultado (S1) igual do esperado (E2)	SUCESSO
5	? Resultado (S1) diferente do esperado (E2)	FALHA
6	? Classificação Impossível	FALHA
7	! Comparar posições (S2) com posições (E3).	
8	? Posições (S2) iguais às Posições (E3)	SUCESSO
9	? Posições (S2) diferentes das Posições (E3)	FALHA

4.2.4. Execução do Teste

À semelhança do que aconteceu no teste anterior este também foi realizado varias vezes, mais uma vez com a intenção de garantir a consistência dos resultados. Este teste foi realizado utilizando vários grupos de itens selecionados através da Interface. Todos os dados foram registados na Tabela 4.6 apresentada abaixo.

Tabela 4.6 - Execução do Teste: Otimização de espaço

Teste	Entrada			Saída		Resultado (Numero de Linha)	
	E1: [Item1,Item2 ,...]	E2: Classificação	E3: Posições	S1: Classificação	S2: Posições	Esperado	Obtido
1	[Item1, Item2, Item3]	Bin Packing Problem	[Item1, Item2], [Item3]	Bin Packing Problem	[Item1, Item2], [Item3]	S1:(4) S2:(4)	S1:(4) S2:(4)
2	[Item3, Item2, Item1]	Bin Packing Problem	[Item3], [Item2, Item1]	Bin Packing Problem	[Item3], [Item2, Item1]	S1:(4) S2:(4)	S1:(4) S2:(4)
3	Sem Itens	Sem Classificação	Sem Posições	Sem Classificação	Sem Posições	S1:(5) S2:(8)	S1:(5) S2:(8)

Após a conclusão dos testes, o veredicto obtido é de SUCESSO. Embora num dos testes não tenha ocorrido classificação ou acomodação, à semelhança do sucedido no teste à aquisição de dados, neste caso não existia informação, ou seja, não haviam sido selecionados quaisquer itens para acomodar sendo assim de esperar este resultado. Já quando foram fornecidas informações a classificação e calculo foram de acordo com o esperado, como podemos observar no teste 1 quando é introduzido um grupo de itens fortemente heterogéneos, [Item1,Item2,Item3], e o resultado desse teste corresponde ao esperado apresentado uma classificação de Bin Packing Problem e um calculo de posições dos itens sendo o item 1 e 2 acomodados no primeiro espaço, [Item1,Item2], e o item 3 no segundo espaço, [Item3]. Assim com base nos resultados obtidos o veredicto obtido é de SUCESSO em ambas as saídas.

4.3. Análise dos Testes efetuados

Perante os resultados dos testes efetuados, os resultados foram os desejados revelando-se coerentes e correspondem ao espectável. Assim da concretização destes testes várias conclusões podem ser retiradas.

No caso do teste referente à aquisição de dados os resultados são muito favoráveis dado que todas as repetições apresentam o veredicto de SUCESSO podendo assim concluir-se que a aquisição de dados por parte da rede de sensores ocorre de forma correta e estes são entregues corretamente ao sistema.

No caso do teste à otimização do espaço os resultados foram igualmente felizes. Com a realização deste teste foi possível observar que todas as classificações foram bem executadas, ou seja, o sistema é capaz de identificar qual o problema correspondente dado o cenário apresentado vindo comprovar que a classificação baseada em prioridades apresentada na árvore de classificação desenvolvida neste trabalho permite uma classificação clara dos problemas de otimização.

O calculo das posições também ocorreu de acordo com o ponderado, apesar da implementação testada não constar de todos os algoritmos, é permitido concluir através dos testes efetuados que este sistema é capaz de determinar a posição que deve ocupar cada item para obter uma otimização do espaço disponível segundo a classificação feita.

4.4. Validação do Cenário Industrial

O piloto industrial C2NET *Metalworking Network* SME's será utilizado como cenário para a validação industrial deste trabalho. O projeto C2NET tem como objetivo melhorar ao sector das pequenas e medias empresas (PMEs) no que diz respeito à competitividade, inovação e adaptabilidade no cenário de redes de parceiros empresarias. Providenciando um “alem fronteiras” e intercambio entre empresas, através da construção de uma rede de empresas que é suportada por esquemas de relacionamento estáveis e paradigmas modernos de cooperação e coordenação de negócios. Isto vai garantir vantagens aos consumidores finais, em termos da redução do tempo de chegada ao mercado e custo dos produtos.

A rede é composta por duas PMEs que trabalham na industria da transformação e que compram aço a fornecedores e transformam essa matéria prima em produtos prontos a serem vendidos no mercado. Na primeira o seu produto especializado são tubos de aço, sendo a segunda especializada na produção de prateleiras de aço. Ambas as empresas procuram reduzir os custos da aquisição de matérias primas aos seus fornecedores tanto no próprio material como no transporte do mesmo.

Tendo em conta que a metodologia de trabalho adotada por estas empresas é do tipo *make to order*, estas empresas não mantêm grandes quantidades de material armazenado, apenas o necessário para a produção. Assim um processo de transporte otimizado revela-se útil para que possam garantir o despacho de produtos e a receção de matérias primas na forma mais eficiente possível, de forma a garantir o baixo stock em armazém.

Contudo mesmo o pouco stock é desejável que este esteja otimizado no espaço existente e consequentemente organizado, pois garantindo isso é garantido também que durante o processo de produção a empresa não será afetada por problemas tais como a falta de stock ou stock perdido (Figura 4.2).

No cenário do C2NET, esta tese pode contribuir para a otimização de espaço nos processos de transporte e armazenamento, tornando possível uma melhor utilização dos recursos disponíveis tais como carrinhas de transporte e armazéns de produtos finalizados ou matérias primas.

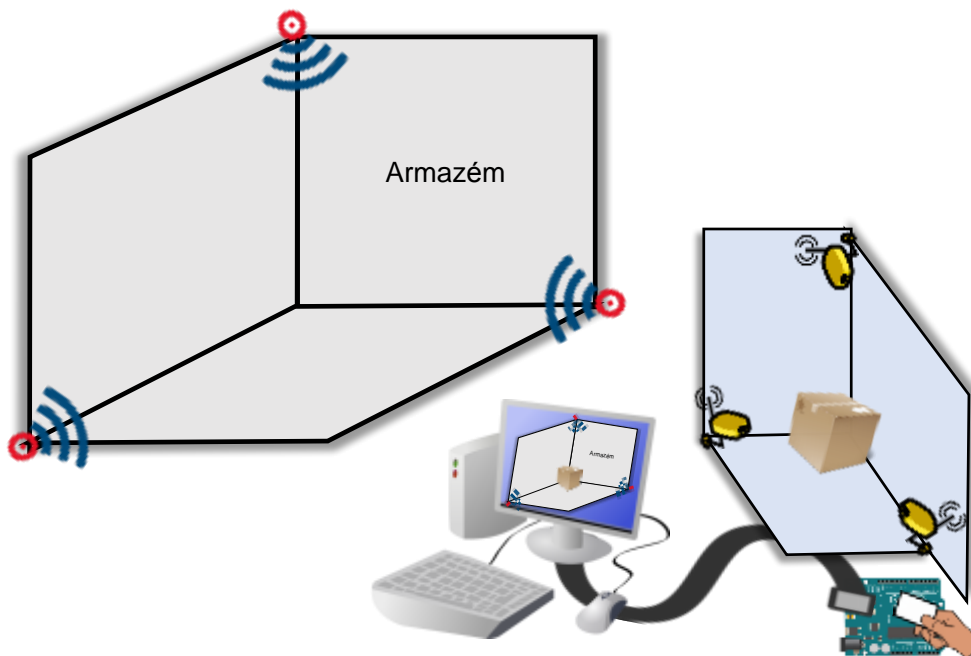


Figura 4.2 - Cenário Industrial - Armazém

Capítulo 5 – Conclusões e Trabalho Futuro

Devido ao crescimento dos mercados mundiais, a transação de bens ocupa uma importante posição nas preocupações dos dias hoje. Empresas de manufatura ou logística, são todos os dias desafiadas a movimentar e armazenar bens de uma forma melhor e mais rápida. Para tal, é necessário às empresas encontrarem formas de melhorar esses processos, ou seja, formas de manter esses bens organizados de forma mais eficiente bem como de os transportar de forma mais rápida sem descurar o plano económico, ou seja, mantendo os custos associados a essas operações o mais baixo possível. Assim, estas necessidades resultaram na seguinte pergunta e consequente hipótese:

Assim, tendo em conta estas necessidades, esta dissertação assumiu como objetivo facilitar o processo de otimização de espaços em cenários de armazenamento e transporte de produtos através da criação de uma árvore de decisão baseada em prioridades, do classificador de cenários e por fim da arquitetura do sistema de otimização inteligente.

Para tal, duas temáticas foram identificadas como partes da solução automática. Estas temáticas são os problemas de otimização de espaço e sistemas para aquisição de dados, de forma a que a sua combinação resulte num sistema capaz de fazer a aquisição automática das informações relevantes, processá-las e devolver a forma mais eficiente de efetuar a acomodação dos bens no espaço disponível.

Através da revisão da literatura referente à temática dos problemas de otimização de espaço, foi possível identificar que muitas definições de problemas de otimização de espaço, estavam fortemente direcionados a um único cenário de aplicação, o que implicava uma caracterização muito específica dos espaços a otimizar, da diversidade de itens e também das dimensões que estes problemas consideram (1D, 2D ou 3D). Apesar disso foi possível identificar uma definição destes problemas onde utilizando como base dois critérios: o tipo de tarefa a desempenhar e a variedade de tipos de itens a acomodar. Assim com estes dois critérios o autor define uma definição mais abrangente que os seus antecessores para a caracterização deste tipo de problemas, que resulta na identificação de seis problemas base: *Identical Item Packing Problem*,

Placement Problem, Knapsack Problem, Open Dimension Problem, Cutting Stock Problem e Bin Packing Problem.

Contudo esta utiliza no seu processo de decisão, fatores tais como lucro e custo, levando a que as caracterizações dela provenientes tenham sempre como tarefa a desempenhar a minimização ou maximização destas características, o que faz com que nem sempre se revele uma opção favorável.

Já na revisão da literatura referente a sistemas de aquisição de dados, foi possível identificar soluções apropriadas às necessidades do objetivo proposto, nomeadamente no âmbito de Cyber Physical Systems. O facto dos Cyber Physical Systems permitirem uma conexão entre o mundo real e o mundo digital, revelou-se uma grande vantagem permitindo assim a ligação a dispositivos IoT para a captura de informação no mundo real. Alguns *middlewares* e arquiteturas foram alvo de estudo para fazer a ponte entre os dispositivos físicos e o sistema digital, neste campo foi estudada a arquitetura definida como referencia, IoT-A, e algumas implementações já disponíveis ou em fase de protótipo, tais como, FIWARE – IoT Stack, OpenIoT e C2NET – IoT HUB.

Assim de forma a corresponder ao objetivo, e tendo em conta a revisão da literatura efetuada, foi desenvolvido um classificador para os problemas de otimização de espaço que através da introdução da característica de prioridade associada aos itens a acomodar, permite ao utilizador revelar quais os itens que tem maior valor, importância ou prioridade. Por exemplo, num cenário em que todo um projeto esteja parado por falta de parafusos, apesar de estes terem um valor monetário reduzido influenciam o desenvolvimento de todo um projeto, tornando-se assim prioritário que estes sejam corretamente acomodados. Outra influencia da introdução da característica perioridade na classificação é o facto de ser deduzível se irá ocorrer uma seleção ou não de itens a acomodar, pois num cenário em que todos os itens tenham a mesma prioridade associada, o utilizador não tem especial preferência por nenhum, levando a que todos eles precisem igualmente de ser transportados, o que num caso de falta de espaço para todos, o espaço para o item deverá ser providenciado, pois segundo o utilizador todos devem ser acomodados. Já no caso de os itens terem prioridades diferentes, é claro que se o espaço não permitir a acomodação de todos, a prioridade deve ser maximizada dentro do espaço disponível

Assim conclui-se que com introdução do critério de prioridade é possível atribuir ao utilizador capacidade para “influenciar” a decisão segundo o que mais lhe for mais conveniente e também que o processo de decisão, deixa de depender apenas da verificação se o espaço existente é suficiente ou não para os itens.

Mais uma vez com intuito de corresponder ao problema a que este trabalho se propõe, foi desenvolvido uma arquitetura composta por um sistema de otimização, um *middleware* e um dispositivo. O sistema é composto por quatro módulos: Interface de Utilizador, Classificador de Cenários, Optimizador de Espaços e Base de Dados. Sendo que as suas funções são no caso da Interface de Utilizador são, permitir ao utilizador selecionar e priorizar os itens que pretende

acomodar, no caso do Classificador de Cenários são, interpretar as características provenientes da IU e classificar o problema, no caso do Optimizador de Espaços são, calcular a posição que cada item deve ocupar dentro do espaço disponível, e no caso da Base de dados são, armazenar a informação referente aos itens e espaço para acomodação e disponibilizar essa informação aos restantes módulos quando necessário.

A arquitetura proposta conta também com um *middleware* e uma rede de sensores. Sendo que o primeiro foi introduzido para permitir a conexão entre o nível aplicacional do sistema e a rede de sensores, e o segundo para permitir a aquisição de informação do mundo real.

É possível também concluir com base nos testes realizados que o processo de aquisição de dados garante a informação com o rigor e no tempo esperado de forma a popular a Base de Dados e consequentemente o sistema. A árvore utilizada no processo de classificação obteve também resultados muito positivos, permitindo uma classificação que respondeu sempre de acordo com o esperado. O processo de cálculo das posições obteve também resultados positivos, apesar deste não ter na sua implementação algoritmos para tratar todos os cenários identificáveis, quando na presença de um cenário Bin Packing Problem este efetua o cálculo segundo o esperado. O que deixa claro que com a introdução dos restantes algoritmos é capaz de calcular as posições em qualquer cenário.

É possível concluir que através da solução proposta e posteriormente implementada, que é possível automatizar o processo de aquisição das informações geométricas dos itens a acomodar e também que a caracterização proposta corresponde aos resultados esperados, na presença de itens regulares. Sendo assim, pode-se concluir que a arquitetura apresentada nesta dissertação, aliada ao método de decisão, também ele apresentado, cumpre o objetivo a que esta se propunha validando assim a Hipótese deste trabalho.

5.1. Trabalho Futuro

De forma a alcançar um Sistema que nas suas plenas funções permita uma otimização efetiva do espaço segundo os itens selecionados para a acomodação, o trabalho apresentado necessita que a implementação do módulo Optimizador seja totalmente concluída por forma a proporcionar os melhores resultados.

Outro trabalho futuro possível para a solução apresentada, seria a introdução da aquisição automática do peso de cada item, através da introdução de um novo sensor. A introdução desta característica, revelar-se-ia como uma mais valia em cenário de transporte, devido ao facto de os meios de transporte, serem limitados no que toca ao peso que transportam, evitando assim que fosse ultrapassado o peso limite. Também seria uma mais valia no cenário de armazenamento pois poderia ajudar a respeitar a resistência de prateleiras nas quais os itens

seriam acomodados, ou, evitando que um item muito pesado fosse acomodado em cima de outro.

A introdução desta característica traria grandes mudanças nos algoritmos de otimização implementados no Optimizador, contudo, durante a revisão da literatura foram encontrados alguns artigos que contemplavam, heurísticas e algoritmos de otimização já tendo em consideração esta característica, o que poderia ser um ótimo ponto de partida para esse trabalho(Kröger, Schwenderling, & Vornberger, 1991; Vijayakumar, Parikh, Scott, Barnes, & Gallimore, 2013).

Capítulo 6 - Bibliografía

- Ashton, K. (2009). That “Internet of Things” Thing - RFID Journal.pdf. Retrieved from <http://rfidjournal.com/articles/view?4986>
- Baheti, R., & Gill, H. (2011). Cyber-physical Systems. *The Impact of Control Technology*, (1), 161--166. <http://doi.org/10.1145/1795194.1795205>
- Baker, B. S., Coffman, Jr., E. G., & Rivest, R. L. (1980). Orthogonal Packings in Two Dimensions. *SIAM Journal on Computing*, 9(4), 846–855. <http://doi.org/10.1137/0209064>
- Bauer, M., Boussard, M., Bui, N., & Carrez, F. (2013). Project Deliverable D1.5 – Final Architectural Reference Model for IoT, (257521), 53–59. Retrieved from http://www.iot-a.eu/public/public-documents/d1.5/at_download/file
- Bauer, M., Kovacs, E., Schülke, A., Ito, N., Criminisi, C., Goix, L. W., & Valla, M. (2010). The context API in the OMA next generation service interface. *2010 14th Int. Conference on Intelligence in Next Generation Networks: “Weaving Applications Into the Network Fabric”, ICIN 2010 - 2nd Int. Workshop on Business Models for Mobile Platforms, BMMP 10*, (September 2016). <http://doi.org/10.1109/ICIN.2010.5640931>
- Complex Event Processing (Perseo) - FIWARE-IOT-Stack. (2015). Retrieved April 3, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/cep/index.html>
- Connector Framework (Cygnus) - FIWARE-IOT-Stack. (2015). Retrieved April 2, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/cygnus/index.html>
- Context Broker (Orion) - FIWARE-IOT-Stack. (2015). Retrieved April 2, 2016, from http://fiware-iot-stack.readthedocs.org/en/latest/context_broker/index.html
- Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1), 1–20. <http://doi.org/10.1016/j.ejor.2016.04.030>
- Device Backend Gateway (IDAS) - FIWARE-IOT-Stack. (2015). Retrieved April 2, 2016, from http://fiware-iot-stack.readthedocs.org/en/latest/device_gateway/index.html
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2), 145–159. [http://doi.org/10.1016/0377-2217\(90\)90350-K](http://doi.org/10.1016/0377-2217(90)90350-K)

- FIWARE-IOT-Stack. (2015a). Retrieved March 30, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/index.html>
- FIWARE-IOT-Stack. (2015b). Retrieved April 1, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/index.html#apis-available>
- FIWARE Components - FIWARE-IOT-Stack. (2015). Retrieved April 2, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/walkthrough/index.html>
- Ghimire, S., Melo, R., Ferreira, J., Agostinho, C., & Goncalves, R. (2015). Continuous Data Collection Framework for Manufacturing Industries. In I. Ciuciu, H. Panetto, C. Debruyne, A. Aubry, P. Bollen, R. Valencia-García, ... F. Ferri (Eds.), *On the Move to Meaningful Internet Systems: OTM 2015 Workshops: Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, EI2N, FBM, INBAST, ISDE, META4eS, and MSC 2015, Rhodes, Greece, October 26-30, 2015. Proceedings* (pp. 29–40). inbook, Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-26138-6_5
- Haims, M. J., & Freeman, H. (1970). A Multistage Solution of the Template-Layout Problem. *IEEE Transactions on Systems Science and Cybernetics*, 6(2), 145–151. <http://doi.org/10.1109/TSSC.1970.300290>
- Höller, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S., & Boyle, D. (2014). *From Machine-To-Machine to the Internet of Things. From Machine-To-Machine to the Internet of Things*. <http://doi.org/10.1016/B978-0-12-407684-6.00005-X>
- Identity Management (IDM) - FIWARE-IOT-Stack. (2015). Retrieved April 3, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/idm/index.html>
- Internet of Things - Architecture — IOT-A: Internet of Things Architecture. (2013). Retrieved April 3, 2016, from <http://www.ietf.org/public>
- ISO/IEC 9646. (n.d.). Retrieved from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=17473.
- Jiang, L., Da Xu, L., Cai, H., Jiang, Z., Bu, F., & Xu, B. (2014). An IoT-Oriented Data Storage Framework in Cloud Computing Platform. *Industrial Informatics, IEEE Transactions on*, 10(2), 1443–1451. <http://doi.org/10.1109/TII.2014.2306384>
- Kang, K., Pang, Z., Xu, L. Da, Ma, L., & Wang, C. (2014). An interactive trust model for application market of the internet of things. *IEEE Transactions on Industrial Informatics*, 10(2), 1516–1526. <http://doi.org/10.1109/TII.2014.2306799>
- Karnouskos, S. (2009). Efficient Sensor Data Inclusion in Enterprise Services. *Datenbank-Spektrum*, 9(28), 5–10. Retrieved from http://diktio.dyndns.org/files/2009_DBSpektrum.pdf
- Kataev, M. Y., Bulysheva, L. A., Emelyanenko, A. A., & Emelyanenko, V. A. (2013). Enterprise systems in Russia: 1992–2012. *Enterprise Information Systems*, 7(2), 169–186. <http://doi.org/10.1080/17517575.2012.745618>
- Khaitan, S. K., & McCalley, J. D. (2015). Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Systems Journal*, 9(2), 350–365. <http://doi.org/10.1109/JSYST.2014.2322503>
- Kopetz, H. (2011). Real-Time Systems. *Real-Time Systems*, 338. <http://doi.org/10.1007/978-1->

- Kröger, B., Schwenderling, P., & Vornberger, O. (1991). Parallel genetic packing of rectangles. In H.-P. Schwefel & R. Männer (Eds.), *Parallel Problem Solving from Nature: 1st Workshop, PPSN I Dortmund, FRG, October 1--3, 1990 Proceedings* (pp. 160–164). inbook, Berlin, Heidelberg: Springer Berlin Heidelberg. <http://doi.org/10.1007/BFb0029747>
- Li, Q., Wang, Z., Li, W., Li, J., Wang, C., & Du, R. (2012). Applications integration in a hybrid cloud computing environment: modelling and platform. *Enterprise Information Systems*, 7575(February 2013), 1–35. <http://doi.org/10.1080/17517575.2012.677479>
- Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2), 243–259. <http://doi.org/10.1007/s10796-014-9492-7>
- Medvedev, A., Zaslavsky, A., Khoruzhnikov, S., & Grudin, V. (2015). Interoperability and Open-Source Solutions for the Internet of Things. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9001(April 2016), 169–182. <http://doi.org/10.1007/978-3-319-16546-2>
- Mitchell, R., & Chen, I. R. (2015). Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems. *IEEE Transactions on Dependable and Secure Computing*, 12(1), 16–30. <http://doi.org/10.1109/TDSC.2014.2312327>
- Onofre, S. (2007). Plataforma para testes de conformidade de sistemas baseados em módulos conceituais STEP, 1–102.
- Partners C2NET. (2015). *Deliverable D3.1 - 1st Data collection from real-world resources: IoT devices*.
- Pedrucci, S., Paulo, L., Nunes, A., Rosa, R. D. A., & Arpini, B. P. (2016). A mathematical model to optimize the volumetric capacity of trucks utilized in the transport of food products, 350–364.
- Ren, L., Zhang, L., Tao, F., Zhang, X. (Luke), Luo, Y., & Zhang, Y. (2012). A methodology towards virtualisation-based high performance simulation platform supporting multidisciplinary design of complex products. *Enterprise Information Systems*, 6(3), 267–290. <http://doi.org/10.1080/17517575.2011.592598>
- Schafersman, S. D. (1997). An Introduction to Science: Scientific Thinking and the Scientific Method. *Geology*, 1–9. Retrieved from <http://www.muohio.edu/~schafesd/documents/intro-to-sci.htmlx>
- Schwerin, P., & Wäscher, G. (1997). The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP. *International Transactions in Operational Research*, 4(5), 377–389. [http://doi.org/10.1016/S0969-6016\(97\)00025-7](http://doi.org/10.1016/S0969-6016(97)00025-7)
- Short Term Historic (STH) - FIWARE-IOT-Stack. (2015). Retrieved April 2, 2016, from <http://fiware-iot-stack.readthedocs.org/en/latest/sth/index.html>
- Silva, S. C. P., Araujo, S. A. de, & Poldi, K. C. (2015). Otimização de problemas integrados na indústria papelreira, 3(lii), 1–7. <http://doi.org/10.5540/03.2015.003.02.0097>
- Sonntag, D. (2014). ERmed - Towards medical multimodal cyber-physical environments. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*

- Intelligence and Lecture Notes in Bioinformatics*) (Vol. 8534 LNAI, pp. 359–370).
<http://doi.org/10.1007/978-3-319-07527-3-34>
- Tao, F., Cheng, Y., Xu, L. Da, Zhang, L., & Li, B. H. (2014). CCIoT-CMfg: Cloud computing and internet of things-based cloud manufacturing service system. *IEEE Transactions on Industrial Informatics*, 10(2), 1435–1442. <http://doi.org/10.1109/TII.2014.2306383>
- Tao, F., Zuo, Y., Xu, L. Da, & Zhang, L. (2014). IoT-Based intelligent perception and access of manufacturing resource toward cloud manufacturing. *IEEE Transactions on Industrial Informatics*, 10(2), 1547–1557. <http://doi.org/10.1109/TII.2014.2306397>
- Tretmans, J. (1992). A formal approach for conformance Testing. *Chemistry &*; Retrieved from <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract>
- Tretmans, J. (2001). An Overview of OSI Conformance Testing. *Methods*, 1–14.
- Viegas, J. L., Vieira, S. M., Henriques, E. M. P., & Sousa, J. M. C. (2015). A Tabu Search Algorithm for the 3D Bin Packing Problem in the Steel Industry. In P. A. Moreira, A. Matos, & G. Veiga (Eds.), *CONTROLO'2014 -- Proceedings of the 11th Portuguese Conference on Automatic Control* (pp. 355–364). inbook, Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-10380-8_34
- Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., & Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224(3), 583–591. <http://doi.org/10.1016/j.ejor.2012.09.010>
- Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109–1130. <http://doi.org/10.1016/j.ejor.2005.12.047>
- Yan, H. H., Wan, J. F., Suo, H., International, C., On, M., & Applied Mechanics, I. (2012). Adaptive Resource Management for Cyber-Physical Systems. In *Mechatronics and Applied Mechanics, Pts 1 and 2* (Vol. 157–158, pp. 747–751). <http://doi.org/10.4028/www.scientific.net/AMM.157-158.747>
- Yesodha, R., & Amudha, T. (2012). A comparative Study on Heuristic Procedures To Solve Bin Packing Problems, 2(6), 37–49. <http://doi.org/10.5121/ijfcst.2012.2603>