Implantação de Precificação Dinâmica em um Estacionamento Inteligente utilizando Agentes*

Alexandre L. L. Mellado[1], Gleifer Vaz Alves[1], Paulo Leitão[2], André Pinz Borges[1]

[1]Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR)
Ponta Grossa – PR – Brasil

[2]Research Centre in Digitalization and Intelligent Robotics (CeDRI)
Instituto Politécnico de Bragança (IPB)
Bragança – Portugal

mellado@alunos.utfpr.edu.br, gleifer@utfpr.edu.br

pleitao@ipb.pt, apborges@utfpr.edu.br

Abstract. *In large cities, parking is a limited resource and drivers spend a lot of time looking for a spot, which causes wasted fuel, congestion and pollution. Here, we develop a method to handle the spot allocation using a dynamic pricing engine. This engine, programmed in Java and implemented in a Multi-Agent System developed in the JaCaMO framework, dynamically changes prices in parking lots. The simulations performed show that a specialized pricing module is better for parking than not using any module.*

Resumo. *Em grandes cidades, estacionamento é um recurso limitado e motoristas gastam muito tempo à procura de uma vaga, o que causa desperdício de combustível, congestionamento e poluição. Neste trabalho é desenvolvido um método para tentar lidar com a alocação de vagas através do uso de um módulo de precificação dinâmica. Esse módulo, programado em Java e implantado em um Sistema Multi-Agente (SMA) desenvolvido no framework JaCaMO, realiza mudanças de preços de estacionamentos dinamicamente. As simulações executadas evidenciam que um módulo de precificação especializado é melhor para o estacionamento do que não usar nenhum módulo.*

1. Introdução

Estacionamentos são recursos essenciais para mobilidade sustentável em áreas urbanas. Esse recurso é limitado em grandes metrópoles, o que leva os motoristas a transitarem por um longo período de tempo para encontrar uma vaga disponível [Shoup 2006]. [Polycarpou et al. 2013] mostra que, durante a procura por vagas de estacionamento, motoristas desperdiçam tempo e combustível, aumentando o congestionamento e a poluição.

Devido a essa limitação de vagas, o preço das vagas pode ser utilizado para orientar as opções de estacionamento dos motoristas de forma dinâmica, buscando melhorar

a taxa de ocupação. Este valor pode ser controlado de forma eficiente em um Estacionamento Inteligente (EI) que é diferenciado por utilizar tecnologias para automatizar e melhorar suas tarefas diárias, como alocação e pagamento de vagas [Di Nocera et al. 2014].

A precificação dinâmica é uma abordagem de alteração do preço de serviços ou mercadorias com base em mudanças no mercado. A precificação pode maximizar a receita de um estacionamento, cobrando um maior valor pelas vagas quando a demanda é mais alta. Isso compensa as potenciais perdas durante um período de baixa demanda. Além disso, o valor reduzido em períodos de baixa demanda pode incentivar os motoristas a utilizarem os recursos dos estacionamentos [Tian et al. 2018].

Além de ser vantajoso para o estacionamento, a precificação dinâmica também é benéfica para os motoristas. Isso é devido às taxas do estacionamento condizerem com o valor real das vagas, em vez de uma cobrança estática arbitrária. Os motoristas que usualmente estacionam durante períodos de baixa demanda ou em áreas com menor demanda perceberão que acabam pagando valores mais baixos pelas vagas [Tian et al. 2018].

O trabalho apresentado neste artigo é uma extensão do projeto MAPS (UTFPR) para gerenciamento de EIs. Alguns resultados já obtidos neste projeto foram: **i.** [Castro et al. 2017] implementam um SMA para alocação de vagas baseado no grau de confiança dos agentes motoristas; **ii.** em [Ducheiko et al. 2018], é apresentado um SMA, onde os agentes negociam por vagas seguindo um modelo aberto e descentralizado; **iii.** [Botelho et al. 2019] propõem a implantação de um EI empregando agentes embarcados.

Neste artigo é proposto um módulo de precificação que pode ser acoplado em um SMA [Wooldridge 2009] de forma que sejam consideradas as variáveis do estacionamento que podem ser modificadas dinamicamente: demanda de vagas e o fator temporal. Esse módulo foi desenvolvido usando como base o SMA, desenvolvido em [Castro et al. 2017]. O SMA foi implementado usando o *framework* [JACAMO 2011], composto por três linguagens: Jason, responsável pela programação dos agentes; Cartago, utilizado para implementar os artefatos do ambiente e Moise, usado na organização dos indivíduos do sistema. JaCaMo foi escolhido por facilitar no desenvolvimento de um SMA ao possuir as linguagens interligadas e porque foi utilizado no SMA usado como base neste trabalho.

O SMA desenvolvimento previamente [Castro et al. 2017] apresenta um mecanismo básico de negociação e alocação de vagas para os agentes motoristas, facilitando o acoplamento de um módulo de precificação dinâmica. Esse SMA compõe a parte de negociação de um Smart Parking. Neste trabalho é desenvolvido um módulo de precificação com viés de identificar o seu impacto sobre a dinâmica de negociações. Almeja-se que utilizando tal módulo, seja possível identificar se a utilização de um módulo de precificação resulta em ganho monetário pra um estacionamento.

O restante do artigo está organizado da seguinte maneira. Na seção 2 são descritos os trabalhos relacionados ao paradigma de precificação dinâmica e projetos MAPS. Na seção 3 tem-se o funcionamento do módulo de precificação. Na seção 4 apresentam-se os resultados das simulações executadas. Na seção 5 tem-se as considerações finais.

2. Trabalhos Relacionados

Nesta seção destacam-se trabalhos que precificação dinâmica e implementação de SMA para EIs.

Em [Chen and Sheldon 2015] é realizada uma medição de como preços dinâmicos afetam a oferta de trabalho de motoristas do Uber, utilizando medições das elasticidades de motoristas e de ofertas positivas. Observa-se que o aumento no preço das viagens diminuem a taxa de pausa, o que leva os motoristas a ajustarem suas programações para dirigir por mais tempo e fornecer mais viagens nos momentos com maior demanda.

Em [Williams 2018] foi criado um modelo de cálculo dinâmico de preços de companhias aéreas, considerando as principais interações entre o padrão de chegada de categorias de consumidores e a capacidade restante sob demanda estocástica. O modelo aumentou os lucros e consumidores do sistema.

Um modelo de precificação dinâmica para reserva de vagas foi apresentado em [Tian et al. 2018]. Apesar do modelo considerar apenas a demanda e número de vagas, este resultou em melhor utilização de vagas e aumento de lucros.

Em [Pierce and Shoup 2013] é revisado o aplicativo *SFpark* que ajusta preços para a aumentar a disponibilidade de vagas. Foram realizados testes com 5000 mudanças de preço, mostrando que o preço médio do medidor caiu 1% durante o primeiro ano. Logo, o *SFpark* ajustou os preços sem aumentá-los, em geral.

Os artigos anteriores mostram exemplos de soluções utilizando precificação dinâmica e sua universalidade na era da informação. Observa-se que a precificação dinâmica pode auxiliar na gestão de alocação de recursos e gerar lucro ao proprietário utilizando estes métodos.

Em relação à trabalhos usando SMA para EIs, [Castro et al. 2017] utilizou o *framework* JaCaMo para alocação de vagas. Este sistema é composto por um agente *manager* e agentes *drivers*. O *manager* é responsável por administrar as vagas que serão alocadas aos *drivers*, de acordo com os respectivos graus de confiança destes agentes. Esse grau de confiança é um valor que mostra o comprometimento do *driver* em relação ao estacionamento. Foram realizadas diversas simulações em cenários empíricos que, em linhas gerais, indicam que um grau de confiança maior resulta em um menor tempo de espera para o agente *driver*.

O trabalho de [Ducheiko et al. 2018] expande o projeto MAPS ao criar um modelo de raciocínio para desenvolver o mecanismo de negociação de vagas de forma descentralizada. Este descreve os protocolos e fórmulas elaboradas para a negociação entre os agentes, e é implementado um SMA utilizando o *framework* JaCaMo, destacando a organização social do sistema através de regras do Moise.

Neste artigo tem-se como objetivo complementar os trabalhos de [Castro et al. 2017] e [Ducheiko et al. 2018] por meio do desenvolvimento de um módulo de precificação genérico, capaz de ser implantado em diferentes tipos de estacionamentos.

3. Desenvolvimento

Este trabalho visa desenvolver um módulo de precificação dinâmico que possa ser usado em qualquer EI. Para implantar esse módulo foi necessário a utilização de um SMA já existente. Esse sistema em questão [Castro et al. 2017] foi modificado para realizar negociações simples entre os agentes *driver* e *manager*.

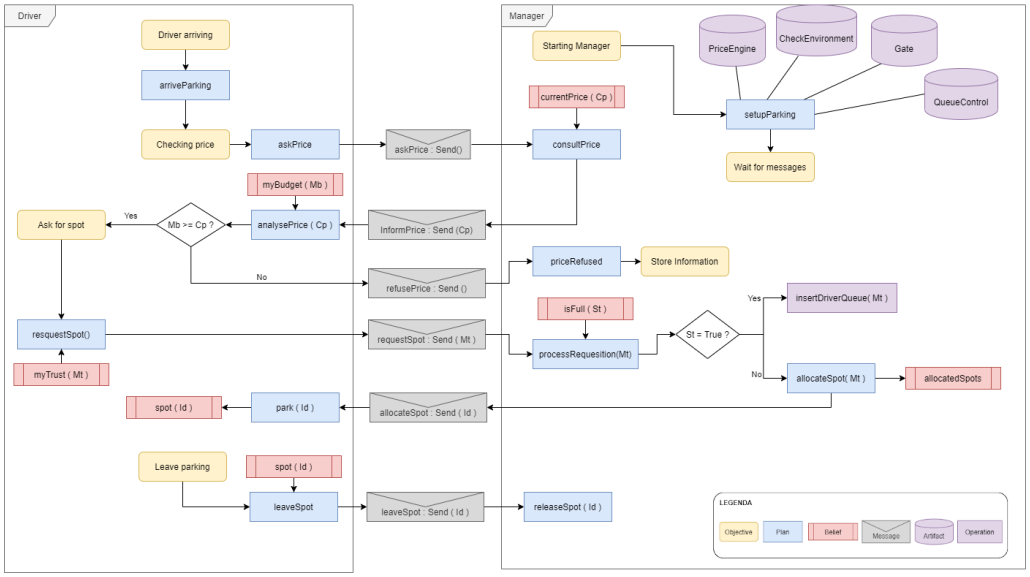

Figura 1. Modelagem do SMA de um EI

A Figura 1 ilustra a modelagem de um EI utilizado como base da implantação do módulo de precificação. Neste SMA centralizado o agente *manager* controla a entrada, alocação e pagamento dos agentes *drivers* interessados em estacionar. Os agentes *driver* encarregados apenas de decidir se aceitam ou não o preço das vagas. A decisão é tomada a partir da comparação entre os créditos disponíveis ao *driver* e o valor das vagas indicado pelo *manager*.

O agente *manager* é um ponto central de gerência de todo o estacionamento. Ele está encarregado de receber, informar e alocar agentes *drivers*. O *manager* também é responsável por decidir quando alterar o preço das vagas de forma dinâmica (após observar o ambiente) ou de forma estática (utilizando planos internos).

O módulo de precificação é encarregado de retornar o novo preço das vagas quando requisitado pelo agente *manager*. O valor é definido a partir das variáveis do ambiente atual e também das regras definidas na configuração do estacionamento (ver Subseção 3.1).

A seguir são descritos os componentes do módulo de precificação: o arquivo de configuração que define as regras seguidas pelo módulo e as operações do módulo que observam a situação do ambiente a fim de atualizar o valor da vaga.

3.1. Regras do Sistema

Parte integral deste módulo de precificação dinâmica encontra-se no arquivo de configuração. Neste sistema foi utilizado um arquivo CSV, contudo o método de processamento do arquivo possibilita utilizar outras extensões de arquivos. No arquivo de configuração são definidas as regras utilizadas para a mudança dos preços das vagas.

A construção deste arquivo deve ser feita da seguinte forma: primeiramente, é criada uma legenda que contém o ID, a *String* e os intervalos das regras. Em seguida, uma tabela contendo as categorias, condições e a porcentagem de mudança do preço.

Na tabela de legenda, a coluna *ID* é usada para identificar o tipo de regras, onde *1* indica uma mudança por demanda; *2* mudança por Clima e *3* mudança por Horário do dia. A coluna *String* representa a palavra usada na tabela de regras para cada *ID*; E a coluna intervalos estabelece os valores usados para escolher a condição de uma categoria.

Na tabela de regras, a primeira coluna contém as categorias que definem o tipo de regra. A segunda coluna contém as condições que especificam uma categoria. A terceira coluna contém os valores de cada combinação categoria *vs.* condição.

Tabela 1. Conf 1

1	**ID**	**String**	**Intervalos**
2	1	Demanda	0-0.3-0.6-0.8-1
3	3	Horario	0-6-12-15-18-24
4			
5	**Categoria**	**Condição**	**Valor**
6	Horario	Madrugada	-0.30
7	Horario	Manha	0.10
8	Horario	MeioDia	0.30
9	Horario	Tarde	0.10
10	Horario	Noite	-0.15
11	Demanda	Baixa	0.0
12	Demanda	Normal	0.15
13	Demanda	Alta	0.30
14	Demanda	Enorme	0.60

Tabela 2. Conf 3

1	**ID**	**String**	**Intervalos**
2	1	Dmd	0-0.4-0.7-1
3	2	Tmp	0-1-2
4	3	Hr	0-12-18-24
5			
6	**Categoria**	**Condição**	**Valor**
7	Tmp	Ensolarado	-0.1
8	Tmp	Nublado	0
9	Tmp	Chuvoso	0.05
10	Dmd	Baixa	0.0
11	Dmd	Media	0.20
12	Dmd	Alta	0.50
13	Hr	Manha	0.0
14	Hr	Tarde	0.05
15	Hr	Noite	-0.05

A tabela de legendas na Tabela 1 e Tabela 2 são utilizadas como meio de identificar quais tipos de regras serão usadas no estacionamento e como elas são definidas. Na Tabela 1 por exemplo, serão utilizadas regras de demanda, clima e horário, enquanto na Tabela 2 serão usados apenas demanda e horário. Os intervalos são usados como meio de conectar a situação do ambiente com as condições disponíveis na tabela de regras. Os valores desses intervalos depende do contexto das regras. Os intervalos de demanda representam porcentagens e precisam conter os valores de 0 a 1 (0 a 100%). Os de clima são identificadores de cada clima observado pelo estacionamento. Os de horário equivalem as horas do dia e devem conter de 00:00 a 24:00 que representam o começo e fim do dia respectivamente.

Cada linha na segunda tabela contém uma regra em que o estacionamento deseja realizar uma mudança do preço. Usando a coluna de intervalos da legenda, no exemplo 1 uma demanda de 65% se encontra entre 0.6 e 1 e logo se encaixa com a condição de *Alta*. A combinação *Demanda Alta*, encontrada na décima quinta linha do exemplo 1, resulta em um aumento de 20% no valor das vagas. Porém, a mesma demanda de 65% encontra-se entre 0.4 e 0.7 na legenda do exemplo 2 e se encaixa na condição *Media*. A combinação *Dmd Media* mantém o valor do preço.

Como o valor base das vagas não muda durante a simulação, o maior e o menor valor possível são determinados pelo arquivo de configuração. Por exemplo, ao usar a Tabela 1 como configuração, o maior preço é encontrado ao ocorrer a combinação de regras *Horario:MeioDia/Demanda:Alta/Temporal:Tempestate*. Usando a mesma tabela, o menor preço ocorre com o uso das regras *Horario:Madrugada/Demanda:Baixa/Temporal:Sol*.

3.2. Variáveis do sistema

Durante a simulação do EI, os valores de algumas variáveis modificam relativamente o funcionamento do sistema. O agente *driver*, quando criado, possui quatro variáveis que determinam suas ações. Essas variáveis recebem valores aleatórios dentro de um intervalo pré-definido. Segue as descrições de suas funções e valores possíveis:

- *timeToArrive:* Determina o tempo em que o agente *driver* irá chegar ao estacionamento após ser criado. Definido entre 0 e 24h (tempo de simulação);
- *timeToSpend:* Determina o tempo em que o agente *driver* irá passar estacionado em sua vaga. Definido entre 30 minutos e 2 dias (tempo de simulação);
- *myTrust:* Variável que determina o valor de confiança usada como meio de priorização de motoristas ao negociar com o agente *manager*. Definido entre 0 e 1000;
- *myBudget:* Valor dos créditos disponíveis ao agente *driver* para requerir uma vaga no estacionamento. Definido entre 5 e 20 créditos.

O agente *manager* possui quatro variáveis. Destas, duas são usadas no gerenciamento do estacionamento e duas são utilizadas para o controle do preço das vagas:

- *nSpotsMax:* Determina o número total de vagas no estacionamento. Este valor é definido estaticamente e altera o espaço total disponível do estacionamento;
- *nSpotsUsed:* Determina a quantidade de vagas ocupadas. A partir desse valor, que é iniciado em zero, e o total de vagas é possível encontrar a ocupação do estacionamento;
- *basePrice:* Valor base do preço das vagas. Esta é a variável usada durante os cálculos de alteração de preço. O seu valor é estático durante a simulação;
- *currentPrice:* Determina o preço atual das vagas. Este valor começa igual ao valor base e é alterado após execuções do modulo de precificação.

3.3. Módulo de Precificação

O módulo de precificação utiliza as regras estabelecidas pelo estacionamento como base para realizar as mudanças de preço. Essas regras determinam quais são as características dinâmicas do ambiente que o estacionamento deseja observar e quanto cada uma delas afeta o preço. Esse módulo é encarregado de receber a circunstância presente do ambiente e modificar o preço das vagas de acordo.

3.3.1. Planos do Agente Driver

O agente *driver* representa ações de um motorista procurando por uma vaga. Neste agente são contidos planos de execução para chegar a um estacionamento, confirmar preço em relação a um orçamento pessoal, estacionar e deixar o estacionamento. Os planos a seguir são parcialmente modificados a partir do sistema desenvolvido em [Castro et al. 2017].

- *setupDriver:* Criação das crenças do agente *driver* e definição de quanto tempo o agente *driver* irá demorar até chegar ao estacionamento. Nesta crenças são contidas: o tempo que o agente deseja ficar no estacionamento, a confiança do agente e o seus créditos disponíveis;

- *arriveParking:* Plano que indica ao sistema que o agente *driver* chegou no estacionamento. É definido como objetivo inicial durante a criação do agente. Ao final do plano é definido como objetivo requisitar o preço das vagas;
- *askPrice:* Agente *driver* envia uma mensagem para o agente *manager* solicitando o preço atual das vagas do estacionamento;
- *confirmPrice:* Plano é iniciado quando o agente *manager* retorna uma mensagem contendo o preço das vagas. Neste plano é realizada a consideração desse valor em relação aos créditos disponíveis do agente. Caso esse valor seja aceito, o agente define como próximo objetivo pedir uma vaga. Caso contrário, o agente envia uma mensagem para o agente *manager* recusando o preço;
- *requestSpot:* Plano é iniciado caso o agente tenha aceitado o preço de uma vaga do estacionamento. Neste plano é enviado uma mensagem ao agente *manager* pedindo por uma vaga disponível;
- *park:* Plano iniciado quando o agente *manager* retorna uma mensagem contendo a vaga alocada. Neste plano o agente *driver* cria uma crença com os dados desta vaga indicada pelo agente *manager*;
- *leaveSpot:* Plano iniciado quando o tempo de estadia no estacionamento termina. Neste plano o agente *driver* remove a crença contendo os dados da sua vaga e envia uma mensagem para o agente *manager* que esta se retirando do estacionamento.

3.3.2. Planos do Agente Manager

O *manager* controla todos os aspectos do EI. Nele é implementada a recepção, alocação e remoção de agentes *drivers*. Ainda são utilizados os artefatos que realizam a verificação do ambiente e alteração do preço das vagas. Os planos de gerenciamento das vagas foram parcialmente modificados a partir do sistema em [Castro et al. 2017]. Os novos planos são aqueles que lidam com a verificação do ambiente e alteração do preço.

- *setupParking:* Inicialização dos artefatos utilizados pelo agente *manager* e criação de suas crenças. Os artefatos em questão são a cancela de entrada do estacionamento, o controle da fila de espera, o sistema de checagem do ambiente, o módulo de precificação e o criador de *log*de eventos.
- *consultPrice:* Plano iniciado após receber uma mensagem do agente *driver* pedindo pelo preço atual das vagas. O preço atual está contido nas crenças do agente manager e uma mensagem com esse valor é retornado ao agente *driver*.
- *processRequisition:* Plano inciado após receber uma requisição de vaga através de uma mensagem vinda do agente *driver*. Neste plano é verificado se existe vaga disponível no estacionamento e desta forma determinar qual é o próximo objetivo do agente manager.
- *priceRefused:* Plano iniciando quando o agente *manager* receber uma mensagem de rejeição do preço de um agente *driver*. Como sistema desenvolvido não possui contra-proposta, o *driver* vai embora do sistema após a rejeição do preço.
- *allocateSpot:* Neste plano é feito uma busca por uma vaga disponível e os dados dessa são alocados ao agente *driver* em questão.
- *insertDriverQueue:* Caso o estacionamento esteja lotado este plano insere o agente *drive* em uma fila de espera utilizando o artefato de controle dessa fila.

Posteriormente esse *driver* irá ser alocado em uma vaga quando uma estiver disponível.

- *releaseSpot:* Plano inciado quando um agente *driver* estacionado decide sair do estacionamento. Esse agente é retirado da vaga e removido do sistema.
- *checkEnvironment:* Neste plano é acionado o artefato de verificação do ambiente e o módulo de precificação. Caso tenha ocorrido alguma mudança no ambiente, prevista no arquivo de configuração (Seção 3.1) o preço das vagas será atualizado de acordo com a nova situação.
- *addPromotion:* Plano que atualiza o preço sem verificar o ambiente. Neste plano o valor das vagas é alterado de acordo com o evento de promoção previamente definido no arquivo de configuração.

3.3.3. Processo de Precificação

Na Simulação implementada o SMA possui artefatos capazes de observar o ambiente e ativar o módulo de precificação, quando ocorrem mudanças.

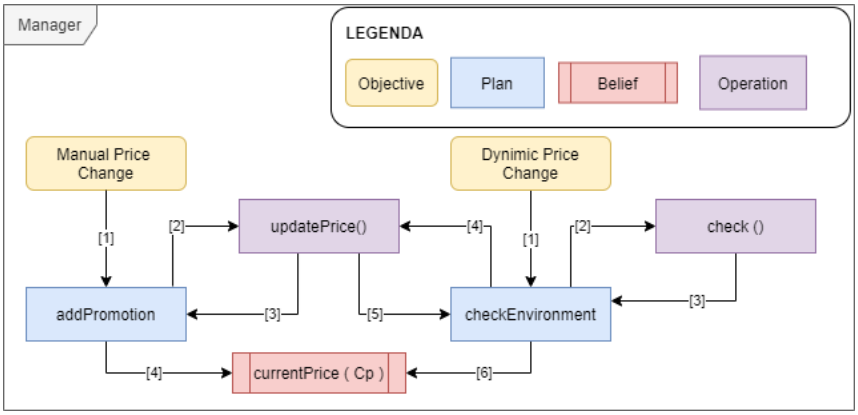

Figura 2. Utilização do módulo de precificação pelo agente *manager*

A Figura 2 ilustra o processo de utilização do módulo de precificação. Em um momento pré-programado no sistema, o agente *manager* inicia o processo de alterar o preço. Dependendo do que causou o início do processo, o *manager* decide entre uma mudança manual ou dinâmica do preço.

Uma mudança manual ocorre quando o agente *manager* deseja mudar o preço a partir de uma crença própria. Na Figura 2, por exemplo, o agente decide acionar o plano *addPromotion* que inclui uma promoção no valor das vagas. Neste caminho é chamada a operação *updatePrice* usando como regra *Evento:Promocao* (ver Tabela 4). Após consultar o arquivo de configuração, essa operação retorna o novo preço e o agente *manager* atualiza o valor das vagas.

Uma mudança dinâmica ocorre quando o agente *manager* deseja verificar se a situação atual do ambiente teve alguma alteração, desde a última checagem. Para isso é iniciado o plano *checkEnvironment*, chamando o artefato que observa o ambiente. A operação *check* do artefato retorna a condição do ambiente e o procedimento utiliza esses dados na operação *updatePrice*. Uma possível condição do ambiente resulta na regra

Demanda:Normal/Temporal:Sol (ver Tabela 4). Essa operação verifica o arquivo de configuração, retorna o novo valor e o agente *manager* atualiza o preço das vagas.

4. Resultados

Aqui são descritos os resultados das simulações do EI. Foram realizadas testes de cenários com e sem a utilização do módulo de precificação.

4.1. Cenários de Testes

Os cenários de teste foram definidos por combinações de números máximos de vagas disponíveis com arquivos de configurações distintos. A partir da combinação entre 50, 100 e 200 vagas, o uso de cinco configurações e ainda desconsiderando o módulo de precificação, no total foram realizadas 18 simulações. Cada cenário foi testado durante 30 dias de simulações. Para cada dia foi criado uma média de 30 *drivers* como meio de simular um número aleatório de clientes do estacionamento. Os cinco cenários desenvolvidos podem ser vistos nas tabelas 1, 2, 3, 4 e 5.

Tabela 3. Conf. 2

ID	String	Intervalos
1	Demanda	0-0.3-0.8-1
3	Horario	0-6-12-18-24
Categoria	**Condição**	**Valor**
Horario	Madrugada	-0.30
Horario	Manha	0.10
Horario	Tarde	0.10
Horario	Noite	-0.15
Demanda	Baixa	0.0
Demanda	Normal	0.15
Demanda	Alta	0.30

Tabela 4. Conf. 4

ID	String	Intervalos
1	Demanda	0-0.3-0.6-1
2	Temporal	0-1-2-3
3	Horario	0-6-12-15-19-24
Categoria	**Condição**	**Valor**
Horario	Madrugada	-0.20
Horario	Manha	0.0
Horario	MeioDia	0.10
Horario	Tarde	0.0
Horario	Noite	-0.10
Evento	Promocao	-0.50
Demanda	Baixa	-0.20
Demanda	Normal	0.0
Demanda	Alta	0.30
Temporal	Sol	-0.1
Temporal	Nublado	0
Temporal	Chuva	0.05
Temporal	Tempestade	0.1

Tabela 5. Conf. 5

ID	String	Intervalos
1	Dmd	0-0.4-0.7-1
Categoria	**Condição**	**Valor**
Dmd	Baixa	0.0
Dmd	Media	0.2
Dmd	Alta	0.50

4.2. Resultados obtidos

Tendo obtido os resultados das simulações foi feita a avaliação do impacto da inclusão de preço dinâmico em relação a ocupação, considerando uma comparação com o sistema sem o módulo. Também foi verificado se o total de créditos recebidos permaneceram equivalentes ou pioraram entre cenários. Cada simulação resulta em um *log* de eventos que contém todas as operações do sistema durante sua execução.

A Figura 3 mostra, como exemplo, dois pedaços de um log de eventos. Os resultados na tabela 6 foram obtidos a partir desses logs de eventos da simulação. Em cada ação dos agentes, foi salvo em um arquivo *csv* contendo as informações e valores dessa ação. Posteriormente foram encontrados os dados mostrados na tabela 6 usando os valores guardados no *log* e calculando as porcentagem de ocupação e valor das vagas ao decorrer do tempo.

As figuras 4, 5 e 6 mostram os gráficos contendo a taxa de ocupação e preço das vagas do estacionamento em cada dia da simulação. Os valores no eixo y são porcentagens para os dados de taxa de ocupação e créditos para os dados de preço atual.

Sector	SpotId	Agent	Description	Date(D/M)	Time(H:M:S)	Rule	CurrentPrice(R$)	OcupationRate(%)
?	?	driver_18	AskedPrice	1/1	1:45:36	?	5	0
?	?	driver_18	RequestedSpot	1/1	1:45:36	?	5	0
B	49	driver_18	SpotAllocated	1/1	1:45:36	?	5	1
?	?	driver_2	AskedPrice	1/1	2:28:52	?	5	1
?	?	driver_2	RequestedSpot	1/1	2:28:52	?	5	1
A	49	driver_2	SpotAllocated	1/1	2:28:52	?	5	2
?	?	driver_1	AskedPrice	1/1	4:49:24	?	5	2
?	?	driver_1	RequestedSpot	1/1	4:49:24	?	5	2
B	48	driver_1	SpotAllocated	1/1	4:49:24	?	5	3
B	39	driver_44	LeavingSpot	2/1	13:1:51	?	3.5	28
?	?	driver_42	AskedPrice	2/1	14:9:25	?	3.5	28
?	?	driver_42	RequestedSpot	2/1	14:9:25	?	3.5	28
B	39	driver_42	SpotAllocated	2/1	14:9:25	?	3.5	29
B	47	driver_15	LeavingSpot	2/1	14:9:25	?	3.5	28
?	?	?	PriceChanged	2/1	15:1:6	Demanda:Normal/ Temporal:Sol	4.5	28
?	?	driver_31	AskedPrice	2/1	15:1:6	?	4.5	28
?	?	driver_31	RequestedSpot	2/1	15:1:6	?	4.5	28
B	47	driver_31	SpotAllocated	2/1	15:1:6	?	4.5	29
B	39	driver_42	LeavingSpot	2/1	15:1:6	?	4.5	28

Figura 3. Exemplo de log de eventos

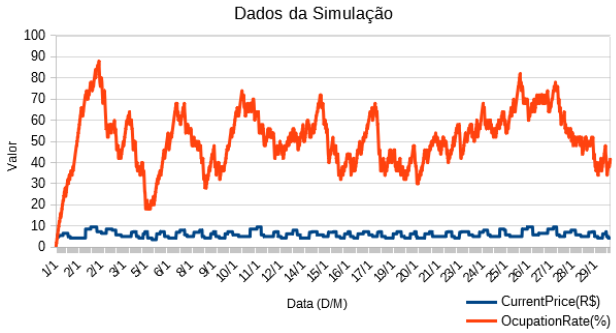

Figura 4. Gráficos de resultados para 50 vagas usando a configuração 1

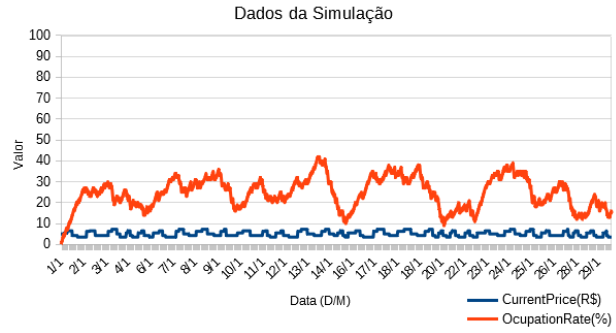

Figura 5. Gráficos de resultados para 100 vagas usando a configuração 1

Os gráficos mostram que os estacionamentos com um maior número de vagas disponíveis tiveram menores taxas de ocupação. Isso ocorrer devido apenas 30 veículos em média entrarem nestes estacionamentos em cada dia. Essa conclusão não é encontrada somente com o uso da configuração 1, como é visto na Tabela 6.

4.3. Discussão dos resultados

A Tabela 6 mostra os resultados obtidos a partir dos *logs* (ver Figura 3) criados durante as simulações realizadas. Os dados de interesse nestes testes foram o crédito total recebido

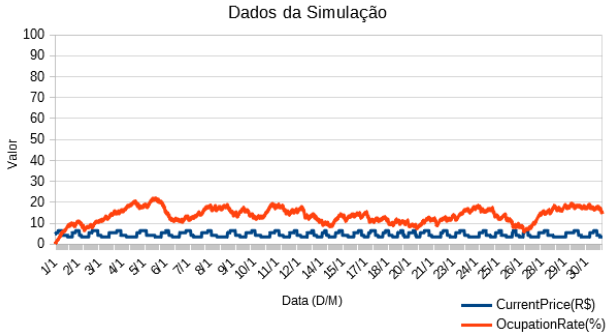

Figura 6. Gráficos de resultados para 200 vagas usando a configuração 1

e a ocupação média durante um mês de simulação.

Tabela 6. Resultados

Módulo	# de Vagas	Crédito Recebido	Ocupação Média
Sem módulo	50 vagas	14220	53,6878%
	100 vagas	14285	23,6682%
	200 vagas	14960	13,4934%
Configuração 1	50 vagas	17747	50,7936%
	100 vagas	14193,75	24,5152%
	200 vagas	13967	13,6244%
Configuração 2	50 vagas	15360,25	49,1860%
	100 vagas	15212,75	26,6203%
	200 vagas	14494,75	13,0428%
Configuração 3	50 vagas	18396,50	47,3752%
	100 vagas	14484,50	24,4134%
	200 vagas	14159,25	14,2786%
Configuração 4	50 vagas	14652,50	44,5488%
	100 vagas	11619,75	24,8263%
	200 vagas	10811,75	13,2768%
Configuração 5	50 vagas	16402,50	45,0777%
	100 vagas	16220	26,0010%
	200 vagas	16085	13,8306%

Em todos os cenários que usaram módulo de precificação, aumentar o número de vagas resultou em uma redução no total de créditos recebidos. Isso é devido às regras de demanda e que o número de veículos não aumenta com o tamanho do estacionamento. Já que os cenários sem módulo não consideram demanda, mais vagas implicam apenas maior número de possíveis clientes e consequentemente mais créditos recebidos.

Apesar da redução dos créditos recebidos em função do aumento do número de vagas, a maioria dos cenários que utilizaram módulos tiveram um valor total maior de créditos recebidos. A clara exceção ocorre na configuração 4 que possui um desconto muito maior em situação de demanda baixa (ver Tabela 4).

Excepcionalmente, quanto maiores as porcentagens de alteração do preço nas regras, maior a diferença nos créditos totais recebidos. Nas configurações dos testes, as regras de demanda possuíram as maiores porcentagens. Especialmente as configurações

1 e 3 (ver Tabela 1 e 2), que tiveram um aumento de 60% e 50% em situações de demanda alta, tiveram os maiores créditos recebidos no cenário de 50 vagas.

É adequado dizer que, se as regras de horário tivessem maiores porcentagens, o número de vagas influenciaria menos o total de créditos recebidos em cada cenário.

5. Conclusão

Em grandes cidades, estacionamentos são fundamentais para a circulação de veículos e redução de congestionamento. A pesquisa de novas tecnologia na área vem a indicar que o uso de sistemas inteligentes é uma alternativa para tentar resolver problemas de trânsito. Neste artigo é apresentada a implementação de um módulo de precificação com o intuito de viabilizar tarifas dinâmicas em um EI. Os resultados da simulação mostram que, a utilização de um módulo de precificação pode acarretar em maiores créditos recebidos como também menores. A ganho total depende de como o módulo foi criado e em onde foram inseridas as maiores porcentagens.

Este trabalho é uma primeira tentativa de desenvolvimento de um sistema de ajustes de preços dinâmicos. Futuramente é necessário a realização de testes explorando uma diversidade maior de cenários, com variáveis mais complexas em relação ao ambiente. Ainda é preciso igualmente aprimorar a simulação para obter resultados mais próximos à realidade. Como trabalho futuro é possível citar uma pesquisa de previsão de ocupação. A partir de mineração de dados armazenados em simulações passadas é possível extrair padrões e realizar recomendações sobre preços futuros das vagas. Esse processo de previsão (na taxa de ocupação) pode ajudar na implementação de um sistema de reserva de vagas com o módulo de precificação dinâmico.

Referências

Botelho, P. W., Borges, A. P., and Alves, G. V. (2019). Proposta de implantação de um sistema ciber-físico para um smart parking baseado em agentes inteligentes.

Castro, L. F. S. D., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a Smart Parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45–55.

Chen, M. K. and Sheldon, M. (2015). Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. *UCLA Anderson. URL: https://www. anderson. ucla. edu*.

Di Nocera, D., Di Napoli, C., and Rossi, S. (2014). A social-aware smart parking application. In *WOA*.

Ducheiko, F. F., André, P. B., and Gleifer, V. A. (2018). Implementação de Modelo de Raciocínio e Protocolo de Negociação para um Estacionamento Inteligente com Mecanismo de Negociação Descentralizado. *Revista Junior de Iniciação Científica em Ciências Exatas e Engenharia*, 1(19):25–32.

JACAMO (2011). The jacamo approach.

Pierce, G. and Shoup, D. (2013). Getting the prices right: an evaluation of pricing parking by demand in san francisco. *Journal of the American Planning Association*, 79(1):67–81.

Polycarpou, E., Lambrinos, L., and Protopapadakis, E. (2013). Smart parking solutions for urban areas. In *2013 IEEE 14th International Symposium on"A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–6. IEEE.

Shoup, D. C. (2006). Cruising for parking. *Transport Policy*, 13(6):479–486.

Tian, Q., Yang, L., Wang, C., and Huang, H.-J. (2018). Dynamic pricing for reservation-based parking system: A revenue management method. *Transport Policy*, 71:36–44.

Williams, K. (2018). Dynamic airline pricing and seat availability.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.