



ALGORITMO PARA OTIMIZAR HEURÍSTICO COMPLEXO ENTRE AS TAREFAS ALGORITHM TO SIMPLIFY HEURISTIC OF COMPLEX PROCESSES BETWEEN TASKS

KOICHI SANOKI; BRÁULIO ALTURAS; ISABEL MACHADO ALEXANDRE
INSTITUTO UNIVERSITÁRIO DE LISBOA;
koichi_sanoki@iscte-iul.pt; braulio.alturas@iscte-iul.pt; isabel.alexandre@iscte-iul.pt

Resumo – Este artigo descreve o algoritmo que otimiza o heurístico complexo de relacionamentos entre tarefas. Essas tarefas são as lógicas computacionais de interação nas aquisições, tratamentos, e disponibilizações de informações. As informações dos dados básicos adquiridos são armazenadas e organizadas em um grupo agregado com indexação para acesso em determinada ordenação que permite relacioná-las com outros grupos de mesma natureza de mídias. A metodologia para a otimização baseia-se na geração de uma indexação única, de tal maneira que os dados se alinhem em uma única forma hierárquica, ou na forma natural da aquisição dos mesmos. Com a composição em uma única indexação o algoritmo descreve a ordem em que as informações são dispostas. Dessa maneira, tais informações são manuseadas por um processo ágil e simplificado em comparação com uma lógica convencional complexa. A escolha de um tipo de heurística, a forma de ordenação da indexação, e o relacionamento entre as tarefas, são a essência para a obtenção do menor percurso no para o acesso e o tratamento de dados. Processo que, pautado nessa indexação, agiliza a disponibilização da informação.

Palavras-chave: Algoritmo Computacional. Heurística. Lógica alternativa. Indexação. Agilização.

Abstract - This article describes the algorithm that optimizes the complex heuristic of relationships between tasks. These tasks are the computational logics of interaction in the acquisition, treatment, and availability of information. The informations of the basic data acquired is stored and organized in an aggregated group with indexing for access in a given ordering that allows it to be related to other groups of the same nature of media. The methodology for optimization is based on the generation of a single index, in such a way that the data are aligned in a single hierarchical form, or in the natural form of the acquisition of the same ones. With the composition in a single indexing, the algorithm describes the order in which the information is arranged. In this way, such information is handled by an agile and simplified process compared to a complex conventional logic. The choice of a heuristic type, the indexing ordering form, and the relationship between the tasks, are the essence to obtain the smallest path in the data access and processing. That process, based on this indexation, expedites the availability of information.

Keywords: Computational Algorithm. Heuristic. Alternative Logic. Indexing.

I. INTRODUÇÃO

O propósito deste artigo é descrever um algoritmo otimizado para o heurístico complexo:

“O método heurístico pressupõe a formulação e a estruturação, de um sistema de ação adequado para cada problema a investigar, o qual é construído durante o próprio processo de investigação.” (W. C. de A. Pereira, 1979).

Com a divisão em várias tarefas simples (N. Facci; C. H. Marroni; L. B. Facci, 2004) e com a utilização do mapa trapezoidal no projeto de mecanização de processos, obter-se-á:

- A análise de um processo manual identificado com uma tarefa de início e final, procurando as funções e as rotinas, e buscando o ponto comum entre essas rotinas. Assim, pode-se ver que existem pontos que estão fora do processo, agilizando então o mesmo;

- A compreensão de que uma vez não entendidas as análises de processos, é gerada uma outra rotina para contornar esse obstáculo. Este projeto mostra que ao invés de contornar esse obstáculo, deve-se achar o espaço livre entre as rotinas;

- A utilização do modelo de Davis (1989) da Figura 2, com seus cinco construtos (Utilidade Percebida - UP; Facilidade de Uso Percebida - FUP; Atitude para Uso - AU; Intenção Comportamental de Uso - IC; e Uso Real - UR) como linha de base para verificar as relações entre as variáveis do modelo, com a aplicação do Mapa Trapezoidal, que no fundo é a Atitude para Uso - AU, e altera sensivelmente o item Facilidade de Uso Percebida - FUP. Portanto essa é a inovação deste projeto em relação aos convencionais.

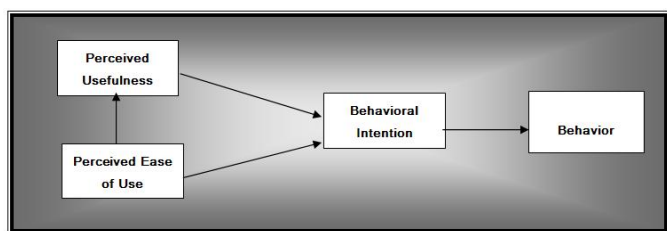
A proposta é que esta pesquisa seja descritiva e exploratória para abordar como o processo de mecanização aceita a colocação de uma metodologia alternativa onde os projetos eram desenvolvidos de forma tradicional.

Neste tema, segundo Kermen e outros (1976), na tarefa entre o início e o término, pode-se construir uma estrutura de dados chamada de “mapa trapezoidal”, que quando utilizada é encontrado o caminho das tarefas sem restrições ou obstáculos que impeçam sua livre execução, e sem que tenha-se que recorrer a alternativas de “ajustes”.

II. METODOLOGIA

Inicialmente, visa-se analisar a utilidade das metodologias quantitativa, qualitativa, e da aplicação do Modelo de Aceitação de Tecnologias, de Davis (1985, 1993), e Davis, Bagozzi e Warshaw (1989), uma vez que este projeto tem como finalidade sua utilização em desenvolvimentos de projetos de mecanização que envolvem as áreas que desenvolvem processos de tarefas de forma virtual. “Davis argumenta que usuários formam tendências motivacionais logo depois de serem expostos a um novo sistema ou projetos e suas respectivas potencialidades. Seria também possível verificar consequências comportamentais como tendências de aceitação ou não. O processo de teste da aceitação consiste em breve demonstração de novos sistemas, utilizando o próprio sistema, ou recursos multimídia de apresentação, seguido da aplicação de uma medida para verificar a motivação para o uso do sistema no trabalho (DAVIS, 1986)”.

Figura 1 - Modelo de Aceitação da Tecnologia (TAM).

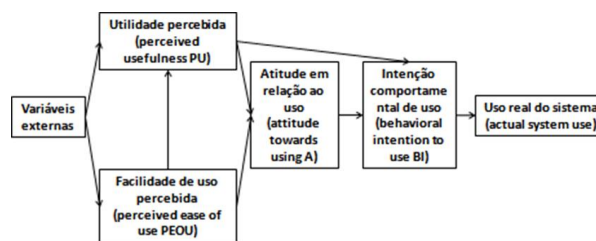


Fonte: Davis (1989).

Essa é a visão geral do Modelo de Aceitação da Tecnologia (TAM), para descrever o processo motivacional entre a tecnologia e o comportamento do usuário. Uma versão mais recente do TAM foi apresentada por Davis, Bagozzi e Warshaw (1989), que aplicaram o modelo em dois estudos, o primeiro desenvolvido a partir de um sistema em uso, e o segundo sobre um sistema que ainda viria a ser utilizado. As medidas de Utilidade Percebida e de Facilidade de Uso Percebida foram refinadas e simplificadas, resultando em dois blocos de seis questões cada, com margem de confiabilidade para Utilidade Percebida de 0,98, e para Facilidade de Uso Percebida, de 0,94 (DAVIS; BAGOZZI; WARSHAW, 1989). Sua estrutura é apresentada na Figura 2.

“No TAM, variáveis externas referem-se às características de sistemas, como menus, ícones e interfaces, ou seja, a tecnologia em si, sobre a qual se está investigando a aceitação. A Utilidade Percebida é compreendida como a possibilidade de conceber vantagem ao desempenho de determinada atividade. A percepção de Facilidade de Uso refere-se à ideia da necessidade de pouco ou nenhum esforço para utilizar determinado sistema” (DAVIS, 1989). Atitude para Uso é o “[...] grau individual de avaliação que vai influenciar a intenção de comportamento” (FISHBEIN; AJZEN, 1975, p. 216). Intenção Comportamental é “[...] uma probabilidade subjetiva individual que determina o comportamento” (FISHBEIN; AJZEN, 1975, p. 288). Portanto, o uso real é uma função de avaliação da frequência de uso, no caso de sistemas já disponibilizados, ou intenção de uso futuro, no caso do desenho de novos sistemas (DAVIS, 1989, p. 989)”.

Figura 2 - Modelo de Aceitação de Tecnologia.



Fonte: Davis, Bagozzi e Warshaw (1989).

Davis, Bagozzi e Warshaw (1989) aplicaram o modelo em dois estudos, o primeiro desenvolvido a partir de um sistema em uso e o segundo sobre um sistema que ainda viria a ser utilizado. As medidas de Utilidade Percebida e de Facilidade de Uso Percebida foram refinadas e simplificadas, resultando em dois blocos de seis questões cada, com margem de confiabilidade para Utilidade Percebida de 0,98, e para Facilidade de Uso Percebida, de 0,94 (DAVIS; BAGOZZI; WARSHAW, 1989).

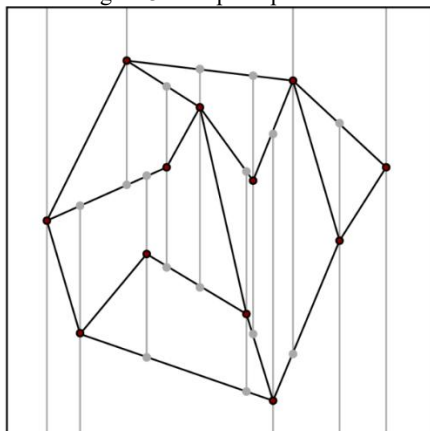
Utilizaram, pois, o modelo de Davis (1989) com seus cinco construtos (Utilidade Percebida - UP; Facilidade de Uso Percebida - FUP; Atitude para Uso - AU; Intenção Comportamental de Uso - IC e Uso Real - UR) como linha de base para verificar as relações entre as variáveis do modelo.

A proposta de Kedem e outros é dividir a tarefa entre o início e o término em um trapézio, para calcular este percurso sem obstáculos entre os dois pontos, o que é possível com essa estrutura. Considerando que uma tarefa “S” qualquer no processo de mecanização com início e final, que não se cruzam, e seja “R” uma caixa de funções de todo o processo contidas na tarefa S, então, o mapa trapezoidal T(S) é obtido traçando-se duas linhas verticais a partir de todos os pontos extremos das funções, uma para cima e outra para baixo, parando em um outra tarefa de S, ou quando tocar em uma das arestas da função R.

O importante é clarificar em que consiste o mapa trapezoidal, visando a compreensão da relação entre o uso do computador (em estudos recentes usado em sentido lato para o uso das tecnologias) por parte dos utilizadores, e o conjunto de variáveis que viabilizavam a aceitação da tecnologia por parte desses mesmos utilizadores.

Esse mapa visa a compreensão da Facilidade de Uso Percebida pelo utilizador, assim como a Utilidade Percebida da tecnologia, relacionando tais conceitos entre si, com a intenção de gerar comportamento, e de fazer uso do sistema e das referidas variáveis externas. A figura 3, mostrando o mapa trapezoidal, foi feita a partir do artigo retirado da fonte <http://upload.wikimedia.org/wikipedia/en/4/43/Trapezoidal_decomposition.png>, que não se encontra mais disponível.

Figura 3 – Mapa trapezoidal.



Fonte: <http://upload.wikimedia.org/wikipedia/en/4/43/Trapezoidal_decomposition.png>.

III. RESULTADOS

Com o mapa trapezoidal obtém-se a representação das tarefas de início e final sem obstáculos, e utiliza-se trapézios que se encontram dentro de polígonos:

3.1 – Variáveis externas

Para calcular o percurso entre o início “s” e o término “t” nesse polígono “P”, basta traçar uma linha reta entre os dois pontos, que estão contidos no trapézio, ou seja, fazendo uma analogia, é como se uma tarefa desse processo estivesse na mesma função.

Agora, caso os pontos estejam em trapézios diferentes, não basta traçar uma linha reta, pois algumas vezes é necessário construir grafos de caminhos livres, fazendo curvas para entrar em determinados trapézios, ou seja, fazendo outra analogia, é como se uma função de uma tarefa utilizasse uma função de outra tarefa, exatamente onde se faria uso de uma terceira função, o que muitas vezes não é feito nessas buscas de “arestas” entre as funções de outras tarefas.

3.2 – Utilidade Percebida

Com esse grafo montado, basta adicionar uma aresta de “s” ao vértice que se encontra no meio do trapézio onde o mesmo “s” está contido, e adicionar uma aresta “t” ao vértice do seu trapézio, buscando a largura ou o algoritmo de DIJKSTRA para achar um caminho possível de “s” a “t”, ou seja, é buscar uma função que muitas vezes não está nos vértices e nem nas arestas, mas sim na largura que pode consistir exatamente na disposição das arestas e vértices de acordo com a disposição das funções dentro das tarefas.

3.3 – Caminhos mínimos

De acordo com os autores Natan e Carlos Eduardo, há de se encontrar o caminho de menor distância euclidiana.

Neste sentido, pode-se enunciar o seguinte lema:

•Lema 1

Qualquer caminho mínimo entre dois pontos “s” e “t”, entre os polígonos em “P”, é um caminho polígamo onde seus vértices são dos polígonos em “P”. Analogamente pode-se dizer que entre as funções de uma tarefa “P” e o

local onde as rotinas são os vértices, então as rotinas da tarefa.

•Prova

Supondo que o caminho “C” não seja um polígono, e seja mínimo, sendo que em obstáculos poligonais existe um vértice “P” em “C” que está contido no espaço livre, e que existe um “E” tal, que o disco “D” com centro “P” e raio “E” esteja totalmente contido em um espaço livre, logo os dois pontos onde o caminho cruza com “D” e conecta-se com eles através de uma linha reta ao invés de passar por “P”, é a heurística de um caminho de comprimento menor que “C”, pois qualquer caminho de extensão mínima tem que ser localmente mínimo também. Dessa forma conclui-se que o caminho tem que ser polígono e tem que excluir a possibilidade de haver pontos contidos no espaço livre. Colocar essa forma em processo mecanizado é dizer que nas tarefas inicial e final existe uma função com uma rotina “P” que está fora do processo da tarefa conhecida, e esse relacionamento está em “D”, tornando assim o caminho “lógico” de relacionamento diferente do conhecido pelas rotinas através das funções que diminuem o caminho para a codificação computacional.

3.4 – Grafo de visibilidade de “P”

Chamado de $Gvis(P^*)$, o grafo de visibilidade de “P”, onde seus vértices são os dos polígonos em $PU\{s,t\} = P^*$, e onde “s” e “t” são os pontos entre os quais se quer a menor distância.

Por definição, os arcos em $Gvis(P^*)$ são entre vértices, que agora incluem “s” e “t”, onde há um caminho em linha reta entre eles, e onde não haja colisão com obstáculos, assim o menor caminho entre “s” e “t” consiste em arcos neste grafo, com o uso do seguinte algoritmo para que se encontre o menor caminho:

○Algoritmo SHORTESTPATH (P,s,t);

○Entrada: um conjunto “P” de polígonos e dois pontos “s” e “t” localizados em algum espaço livre;

○Saída: O menor caminho sem colisão entre “s” e “t”;

○ $G(vis) \leftarrow VISIBILITY\ GRAPH(P \cup \{s,t\})$

○Qualquer ij pertence $G(vis)$, custo $(i, j) \leftarrow$ a distância euclidiana do segmento ij .

○Uso do algoritmo de *Dijkstra* para calcular o caminho mais curto entre “s” e “t”.

IV. CONCLUSÃO

A técnica utilizada nesta pesquisa está de acordo com a divisão em várias tarefas simples e com a utilização do mapa trapezoidal no projeto de mecanização de processos, identificado com uma tarefa de início e final, determinando as funções e as rotinas, e obtendo o ponto comum entre essas rotinas.

A compreensão de que uma vez não entendidas as análises de processos, é gerada uma outra rotina para

contornar esse obstáculo mostrando que ao invés de contornar esse obstáculo, acha-se o espaço livre entre as rotinas.

V. REFERÊNCIAS BIBLIOGRÁFICAS

D. Rozoff. *Heuristic - The Accounting Review*, 39(3), 768-769. 1964. Disponível em <<http://www.jstor.org/stable/242477>>.

E. K. Burke; M. Gendreau; M. Hyde; G. Kendall; G. Ochoa; E. Özcan & R. Qu (2013). *Hyper-heuristics: a survey of the state of the art*, Journal of the Operational Research Society, 64:12, 1695-1724, DOI:10.1057/jors.2013.71. Disponível em <<https://www.jstor.org/journal/joperresoci?refreqid=excelsior%3Af0d13f36775f8a16c737b5dce4e1cb75>>.

F. D. Davis. *Perceived usefulness, perceived ease of use and user acceptance of information technology*. Mis Quarterly, v. 13, n. 3, p. 319-340, 1989.

F. D. Davis, R. P. Bagozzi, P. R. Warshaw. *User Acceptance of Computer Technology: A Comparison of Two Theoretical Models*. Management Science, v. 35, n. 8, p. 982-1003, 1989.

J. Leung. A New Graph-Theoretic Heuristic for Facility Layout. *Management Science*, 38(4), 594-605. 1992. Disponível em <<http://www.jstor.org/stable/2632437>>.

J. O'Rourke. *Computational Geometry in C (Cambridge Tracts in Theoretical Computer Science)* ISBN 10: 0521640105 ISBN 13: 9780521640107 Publisher: Cambridge University Press, 1998.

K. Kedem; R. Livne; J. Pach; M. Sharir. *On the union of Jordan regions and collision-free translation motion amidst polygonal obstacles*, Discrete Comput. Geom. 1 (1986), 5971.

K. Kedem; M. Sharir, *An efficient algorithm for planning collision-free translation motion of convex polygon object in 2-dimensional space amidst polygonal obstacles*. Proceedings of the 1st Annual Symp. Comp. Geom. (1985), 7580.

K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. The University of Chicago. 1993.

K. Sheibani. A fuzzy greedy heuristic for permutation flow-shop scheduling. *The Journal of the Operational Research Society*, 61(5), 813-818. 2010. Disponível em <<http://www.jstor.org/stable/40608242>>.

M. de Berg; M. van Kreveld; M. Overmars; O. Schwarzkopf, *Computational Geometry: Algorithms and applications*, ISBN 978-3-540-77974-2, New York, 2008.

M. Fishbein; I. Ajzen. *Belief, attitude, intention, and behavior: an introduction to theory and research*. Boston (MA): Addison-Wesley, 1975.

M. Libicki; L. Ablon; T. Webb. A Heuristic Cybersecurity Model. In *The Defender's Dilemma: Charting a Course Toward Cybersecurity* (pp. 61-98). Santa Monica, (2015). Calif.: RAND Corporation. Disponível em <<http://www.jstor.org/stable/10.7249/j.ctt15r3x78.13>>.

M. Nagano; J. Moccasin. A High Quality. Solution Constructive Heuristic for Flow Shop Sequencing. *The Journal of the Operational Research Society*, 53(12), 1374-1379. 2002. Disponível em <<http://www.jstor.org/stable/822727>>.

N. Facci; C. H. Marroni; L. B. Facci. Análise do relacionamento sistêmico entre o processo decisório e o Sistema de Informação Gerencial: Um Estudo de Caso em Frigorífico. *Revista Cesumar*, v. 9, n. 1. 2004.

N. C. Lima; C. E. Ferreira. *Algoritmos e estrutura de dados para problemas de deslocamento no plano*. IV Simpósio de Iniciação Científica e Jornadas de Iniciação Científica. Rio de Janeiro, 2008.

R. Zamani; L. Y. Shue. Solving Project Scheduling Problems with a Heuristic Learning Algorithm. *The Journal of the Operational Research Society*, 49(7), 709-716. 1998. doi:10.2307/3010241.

U. Aickelin; A. Clark. Guest Editorial: Heuristic optimisation. *The Journal of the Operational Research Society*, 62(2), 251-252. 2011. Disponível em <<http://www.jstor.org/stable/41058898>>.

W. C. de A, Pereira. O método heurístico em pesquisa. *Journal Sul-Americano Medicina* 1, n.1. p. 21-27, 1979.

VI. COPYRIGHT

Direitos autorais: O(s) autor(es) é(são) o(s) único(s) responsável(is) pelo material incluído no artigo.