How to Efficiently Solve Internet Shopping Optimization Problem with Price Sensitive Discounts?

Jedrzej Musial[1], Johnatan E. Pecero[2], Mario C. Lopez[3], Hector J. Fraire[3], Pascal Bouvry[2] and Jacek Blazewicz[1]

[1]*Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznan, Poland*

[2]*Computer Science and Communications Research Unit, University of Luxembourg, 6 rue Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg*

[3]*Instituto Tecnológico de Ciudad Madero, Ciudad Madero, Mexico*

Keywords: Optimization, e-Commerce, Internet Shopping, Applications, Algorithms, Computational Complexity.

Abstract: In this paper we deal with the Internet Shopping Optimization Problem. An extended model that includes price sensitive discounts is considered. A set of algorithms to solve the Internet Shopping Optimization Problem with Price Sensitivity Discounts (ISOPwD) is introduced. The algorithms are designed to consider a different solution quality regarding computational time and results close to the optimum solution. Simulations based on real world data assess the new set of heuristics. The results of the proposed algorithms were compared with the optimal solutions, computed by a branch and bound algorithm. The scalability is evaluated by increasing the problem sizes. Computational experiments are performed and their results are carefully analyzed and discussed. The paper should be perceived as a work in progress - position paper.

1 INTRODUCTION

The Internet gives access to a huge marketplace. By simply clicking or by touching a screen customers can buy any product online. Electronic commerce (e-commerce) enables consumers to shop at thousands of online stores and to pay for their purchases without leaving the comfort of their homes (Timmers, 1998). Technology (e.g., Cloud Computing, mobile devices - smartphones, tablets) is present into daily business and administrative operations. Already since 2005 one third of European companies used e-commerce in business procurement, logistics, finance and product development (Luxembourg-Embassy-Copenhagen, 2005).

Internet shopping, fitting into a business-to-consumer subcategory, becomes more and more popular. Products available in online stores are often cheaper than those offered by regular local retailers, and a wide choice of offers is available just a click away from the customer (Lee, 1998). Based on outstanding logistics, the delivery can usually be operated within 48 hours or less. A crucial aspect of online shopping is the time spent on comparing offers. Customers often need to take into account that shipping cost are charged, so that it is a good idea to group purchased products into sets and buy them from small number of retailers to minimize these delivery costs. Automating such decisions requires three elements: information about the product availability, price lists, and finally a specialized analytical tool that could find the minimal subset of shops where all the products from the customer's shopping list could be bought at the lowest price (Musial, 2012).

We investigate an extended version of the Internet Shopping Optimization Problem (Blazewicz et al., 2010) which additionally considers price sensitive discounts (Blazewicz et al., 2014). A new set of heuristic approaches to solve the problem was introduced. The set of heuristics is composed of a new lightweight metaheuristic based on a cellular optimization process, a new greedy algorithm and two state-of-the-art greedy algorithms. We have designed different heuristics to consider a different solution quality regarding computational time and results close to the optimum solution. Moreover, the performance of the proposed heuristics to the optimal values was tested. The optimal solutions for small problem instances are computed using a branch and bound algorithm. To evaluate the scalability of the heuristics the problem size was increased.

Musial J., Pecero J., Lopez M., Fraire H., Bouvry P. and Blazewicz J..
How to Efficiently Solve Internet Shopping Optimization Problem with Price Sensitive Discounts?.
DOI: 10.5220/0005112602090215
In *Proceedings of the 11th International Conference on e-Business* (ICE-B-2014), pages 209-215
ISBN: 978-989-758-043-7

2 RELATED WORK

Motivated by the problem of buying multiple products from different e-commerce web sites, Blazewicz et al. (Blazewicz et al., 2010) modeled Internet shopping as an optimization problem, where a customer wants to buy a list of products from online stores at the minimum final price. The authors showed that the problem is NP-hard in the strong sense and designed a set of polynomial time algorithms for special cases of the problem. During previous research different versions (specializations) of the Internet Shopping Optimization Problem were examined (Wojciechowski and Musial, 2010; Blazewicz et al., 2014). For example, due to NP-hardness of the optimization problem, Wojciechowski and Musial (Wojciechowski and Musial, 2010) designed a heuristic solution to optimize the shopping basket and evaluate it for the customer basket optimization problem to make it applicable for solving complex shopping cart optimization in online applications. Moreover, it is proven that the problem is not approximable in polynomial time (Blazewicz et al., 2010). The archetype of the presented problem was a web-based customer assistance system dedicated to pharmacy shopping that helps customers find shops in a geographically defined range where the entire shopping list could be realized at the best total price (Wojciechowski and Musial, 2009).

It is worth noticing that there are some similarities between the Internet Shopping Optimization Problem and the well-known Facility Location Problem (FLP) (Revelle et al., 2008). The main characteristics of the FLP are space, the metric, given customer locations and given or not given positions for facility locations. A traditional FLP is to open a number of facilities in arbitrary positions of the space (continuous problem) or in a subset of given positions (discrete problem), and to assign customers to the opened facilities so that the sum of opening costs and costs related to the distances between customer locations and their corresponding facility locations is minimized.

Discussions of FLPs can be found in (Krarup et al., 2002; Eiselt and Sandblom, 2004; Melo et al., 2009; Iyigun and Ben-Israel, 2010). The traditional discrete FLP is NP-hard (Garey and Johnson, 1979) in the strong sense. Note, however, that the general problem ISOP with price sensitive discounts cannot be treated as a traditional discrete FLP since there is no evident motivation for a discount on the cumulative cost in the sense of distances. It can be noticed that the FLP problem and problem ISOP are not each other's sub-cases, while the traditional discrete FLP is a special case of any of these problems.

The problem of the Internet Shopping with Price Sensitive Discounts (let us call it ISOPwD) has been introduced in (Blazewicz et al., 2014). Computational complexity of various special cases is established. Properties of optimal solutions are proved and polynomial time and exponential time solution algorithms based on these properties are designed. The authors presented two greedy heuristics. In this paper, we consider the same problem. We provide a working model for the experimental results, which is close to real Internet shopping conditions. For that, we prepared a model on the basis of data from the online book industry reported in (Clay et al., 2001) (see subsection 4.1).

3 PROPOSED ALGORITHMS

The ISOPwD is strongly NP-hard. Moreover, to our best knowledge it cannot be a goal of reduction from any one of the known problems. Therefore, it is apparent right to propose heuristic solution, simple efficient greedy algorithms (Blazewicz et al., 2014; Cormen et al., 2001) that use local knowledge and do not allow any backtracking for efficiency purpose. It is worth noticing that the greedy algorithm does not always yield optimal solutions. However, it could provide an optimal or close to optimal solution using much less resources and time than other optimal working algorithms (i.e., full scan, branch&bound). More sophisticated algorithms should provide better quality solutions. Therefore, we propose very efficient pseudo-parallel optimization approach algorithm. State-of-the-art minimization scheduling driven algorithm was used as a competitor to the previous ones.

3.1 Greedy Algorithm - *GREEDY*

In the first heuristic for the ISOPwD, denoted as GREEDY, products are considered in a certain order. The algorithm is run for various product orders and the best solution found is presented to the customer. Let us consider that the products are sorted in an ascending order $1,\ldots,n$. Values of the total delivery and the standard price for all shops are initially set as $T_i = d_i$, $i = 1,\ldots,m$. In iteration j of algorithm GREEDY, product j is selected in its eligible shop $i \in M_j$ with minimum value $f_i(T_i + p_{ij})$, and the corresponding T_i-value is re-set: $T_i := T_i + p_{ij}$.
Discounting function $f_i(T_i, j)$ returns value of product j after applying discount for shop i.

We observed that algorithm GREEDY demonstrates very good performance on the experimental data. The first experimental evaluation of algorithm

GREEDY can be found in (Wojciechowski and Musial, 2010) and (Blazewicz et al., 2014).

3.2 Algorithm with Forecasting - *Forecasting*

Observed weak points of algorithm GREEDY (poor results for a very specific situation) led to the creation of a new, upgraded version. The local step choice analysis is more complicated than in the basic algorithm GREEDY. The forecasting method is looking for a step ahead (Błazewicz and Musiał, 2011). Therefore, technically this algorithm is not a strict greedy algorithm. Sometimes it proposes a current solution which is not optimal for the current step (local solution), but for a better overall solution in hope of providing an optimal global solution. The main idea is to check the situation one step ahead (forecasting bad situations).

From the first step to the penultimate, the algorithm calculates the "choosing factor" to pick an eligible shop i for product j. Instead for looking for a local optimum it looks one step ahead (which prevents occurring a bad case) by calculating a choosing factor as $f_i(T_i, \frac{p_{ij}+p_{ij+1}+d_i}{2})$ for every shop i, and actual product j, as well as following product $j+1$. Then it pick a shop i with the lowest calculated value. In each following step next product j is taken into account, $j = j+1$. The last step of the algorithm works in a different way (forecast could not work beyond the set of products j). The last product is selected in its eligible shop $i \in M_j$ with minimum value $f_i(T_i + p_{ij})$.

Discounting function $f_i(T_i, j)$ returns value of product j after applying the discount for shop i.

3.3 Cellular Processing Algorithm - *Cellular*

The cellular processing based algorithm is a new pseudo-parallel optimization approach (Terán-Villanueva et al., 2013). It includes multiple processing cells that explore different regions of the search space. Each processing cell can be implemented using population or search based heuristics or a hybridization of them. The main idea and the principle of the algorithm are to split a sequential algorithm into several pseudo-parallel processing (i.e. cell) modules, so that each cell can explore different regions of the search space. The main feature of the new approach is that the iterative verification of the stagnation conditions prevents wasting time on unnecessary tasks.

The components of the algorithm are a *pool* of candidate solutions, generated either by a constructive or a random algorithm and a *cell* set that is simple, independent, self contained and applied to work with the subset of candidate solutions that were given to solve. This process continues until the cells stall all the solutions in their local optimum. After that, the solutions return to the pool, and the cells share information with each other in order to escape from the local optimum and continue the search for the global optimum.

In this work we prepared a special designed and implementation of the cellular processing algorithm idea. The candidate solutions were generated at random. An Iterated Local Search algorithm (ILS) was designed as the core of the cells. This choice was made due to the simplicity and high configurability of this structure, which allows it to be highly scalable and to run in a variety of hardware configurations. Moreover, the ILS algorithm is a trajectory-based metaheuristic that can be seen as a straightforward, yet powerful technique for extending simple local search algorithms.

The algorithm starts off by generating an initial solution. Then, a local search process is applied to the candidate solution. After that, wallowing iteration based approach, it seeks to improve the solutions from one iteration to the next. At each iteration, a perturbation of the obtained local optimum is carried out. The perturbation mechanism introduces a modification to a given candidate solution to allow the search process to escape from a local optimum. A local search is applied to the perturbed solution. The new solution is then evaluated and accepted as the new current solution under some conditions. The algorithm finishes when the termination condition is met.

Algorithm is still under upgrade. Its complicated nature results in updates after completion of each set of experiments.

Cellular Processing Algorithm should be perceived as a new, strong contribution to this paper. The idea is to use a new pseudo-parallel optimization approach. However, all the algorithm development process was created from the basis. All steps in development process were carefully analyzed according to the problem nature.

3.4 Minimum-Minimum Algorithm - *MinMin*

The min-min algorithm is a heuristic that is used primarily in the scheduling of tasks or processes (Ibarra and Kim, 1977). As noted in (Wu et al., 2000), the min-min algorithm is a fast method that offers good performance, but as it schedules those tasks or processes with minimum cost first, it may result in an imbalanced solution.

In (Braun et al., 2001) the process of the heuristic min-min is described in the context of the scheduling tasks onto heterogeneous distributed computing systems. This method was adapted for the Internet Shopping Optimization Problem for the selection of a list of products in a set of stores.

The process begins with the search of the product in the list of unassigned products N, which minimizes the total cost TC in the shopping cart among the different stores M, given the cost of the product, p_{ij} plus delivery cost, d_i. In the case of a tie, the product with the lower delivery cost is selected, and if both stores have the same delivery cost, the product is chosen randomly.

Once assigned, the total cost of the shopping cart is updated with the selected product, and it is removed from the unassigned product list N. The process continues until the unassigned product list is empty.

Minimum-Minimum algorithm should be perceived as partly new contribution to this paper. Algorithm was developed according to the literature and then tuned to the ISOPwD.

3.5 Branch and Bound Algorithm - *BB*

To calculate the optimal solution, we designed a Branch and Bound Algorithm (Land and Doig, 1960).

The algorithm starts off by calculating an upper bound (UB) employing the solution given by the Cell Processing Algorithm and proceeds to branch the first level of the search tree in the stack.
Subsequently, the algorithm pops the top element of the stack and evaluates the objective value it would have if it were part of the current solution. If the partial solution exceeds the limit given by the upper bound (UB), the current branch is fathomed. Otherwise, if it were not a leaf of the tree, the algorithm would pile up the following elements within the stack. If it were a leaf, it would mean that the current solution is better than the best global solution found so far. Consequently it would update the upper bound with the new value found.

This process continues as long as there are elements in the stack, which means that the whole search tree has been explored. Founded upper bound is now considered as the optimal solution of the instance being evaluated.

Branch and Bound algorithm should be perceived as partly new contribution to this paper. Algorithm was developed according to the literature and then tuned to the ISOPwD.

4 COMPUTATIONAL EXPERIMENTS RESULTS

Computational experiments were performed and divided into two groups (due to the computational complexity time) - a set of experiments including Branch and Bound exact algorithm, and a set of experiments without optimal solution as a comparison of all heuristics to evaluate scalability issues by increasing instances' size. Very first experimental results from the first group will be presented within this section. Full results will be presented in the final version of the article.

Algorithms were compared using three metrics: an approximation factor, the running time spent by each heuristic to compute a solution, and the dispersion analysis based on the standard deviation. The approximation factor of a heuristic is defined as $\rho = \frac{F(X)}{F(X)^*}$, where $F(X)$ represents the solution found by a heuristic and $F(X)^*$ denotes the optimal solution.

4.1 World Working Model and Instances Generator

A challenging step in experimental research was to create a model, which should be as close to real Internet shopping conditions as possible. We studied the relationship between the competitive structure, advertising, prices and price dispersion over Internet stores. As a group of representative products to be taken into account in our computational experiment we chose books, because of their wide choice in virtual (Internet) stores and frequency of purchase through this kind of shopping channel. We adopted some information and computational results from Clay et al. (Clay et al., 2001) for our model.

In the computational experiments we assume that number of stores $m \in \{20, 40\}$, and number of products in our shopping cart $n \in \{2, 3, \ldots, 10, 15, \ldots, 100\}$. It is assumed that each bookstore has all the required books. For each pair (n, m), 100 instances were generated. In each instance, the following values were randomly generated for all i (shop index) and j (product index) in the corresponding ranges. Delivery price: $d_i \in \{5, 10, 15, 20, 25, 30\}$, *publisher's recommended price* of book j: $r_j \in \{5, 10, 15, 20, 25\}$, and price of book j in bookstore i: $p_{ij} \in [a_{ij}, b_{ij}]$, where $a_{ij} \geq 0.69 r_j$, $b_{ij} \leq 1.47 r_j$, and the structure of intervals $[a_{ij}, b_{ij}]$ follow from information from Clay et al. (Clay et al., 2001) and observations on the biggest Polish Internet bookstores. Discounting functions were prepared for every shop.

During full paper submission it is planed to prepare open access online instance generator that was used during computation experiments.

4.2 A Set of Experiments including Branch and Bound Algorithm

The first group of experiments is the one in which the optimal solutions obtained by the exact BB algorithm were compared to two state-of-the-art heuristic algorithms: GREEDY, FORECASTING, and two newly developed algorithms: CELLULAR and MINMIN (for the ISOPWD). In these examples, $m \in \{20\}$, $n \in \{2,3,4,5,6,7,8,9,10\}$, and discounts follow the proposed discounting function. For each pair (n,m), 100 instances using the information in subsection 4.1.

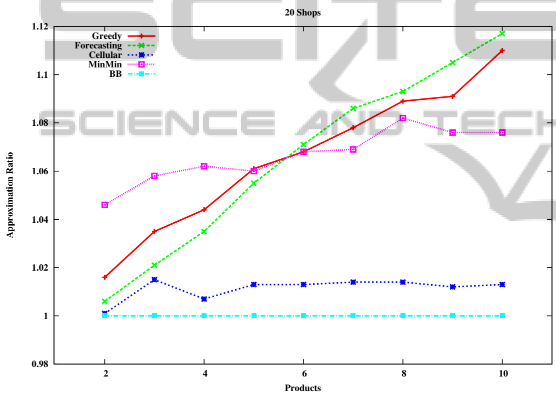

Figure 1: Algorithm results comparison - experiment with 20 shops (including the optimal solution).

After some very first computational experiment we one can notice that among all heuristics, CELLULAR provides the best quality of solutions (closest to the optimum). In the worst case, the solution proposed by this algorithm was merely 1.47% more expensive than the cheapest one. For many instances CELLULAR algorithm computes the optimal solution. It is worth noticing that the algorithm is very stable as regards the quality of solutions (for most cases the solution is between 1.24% - 1.47% worse than optimum).
Both GREEDY and FORECASTING algorithms provide similar quality of solutions (for a lower number of products n, the latter was better and for a higher number of products $n > 5$ the former was better). Moreover, it is easily noticeable that the quality of solution degrades with the increasing number of products n.
The last heuristic algorithm - MINMIN provides the worst solutions for a lower number of products. However, the algorithm is quite stable in quality, therefore for a bigger number of products it provides better solutions than GREEDY and FORECASTING.
If one is looking solely for the quality of solutions the undisputed leader among heuristic algorithms is CELLULAR. The following paragraph will address running time, which is also a very important factor.

For a low number of n products algorithm MINMIN is the fastest. For a bigger number of products, algorithm GREEDY is the fastest. Differences between all algorithms are very significant. The following example illustrates these differences. Computation time for an instance of $n = 5$ products (from $m = 20$ shops) vary from 0.0047 ms for MINMIN) across 0.0085 ms for GREEDY, 0.0205 ms for FORECASTING, 0.3795 for CELLULAR to the 2.5998 ms for BB (computes 556 times longer than MINMIN). For a maximum number of products ($n = 10$) computation times are as follows: 0.0072 ms for GREEDY across 0.0083 ms for MINMIN, 0.0132 ms for FORECASTING, 0.2183 for CELLULAR to the 707.4174 ms for BB (98k times longer than GREEDY).
Computation times for BB algorithm grow exponentially and it was impossible to prepare experiments for a bigger number of products. On the other hand, all heuristics are very fast so the idea is to further test its quality and computational times for scalability issues by increasing the number of products n - let us follow the next subsection.

5 CONCLUSIONS AND CONTRIBUTIONS

In this position paper, the Internet Shopping Optimization Problem including delivery discounts was addressed. For the practical application, a working model as close to real Internet shopping conditions as possible was created.

The main contributions of current research are as follows. A new set of algorithms is presented. Cellular Processing Algorithm should be perceived as a new, strong contribution to this paper. The idea is to use a new pseudo-parallel optimization approach. However, all the algorithm development process was created from the basis. All steps in development process were carefully analyzed according to the problem nature. Another Minimum-Minimum algorithm should be perceived as partly new contribution to this paper. Algorithm was developed according to the literature and then tuned to the ISOPWD. Subsequent Branch and Bound algorithm should be perceived as partly new contribution to this paper. Algorithm was developed according to the literature and then tuned to the ISOPWD. Another two presented algorithms

Greedy and Forecasting were known from the literature. They were perfectly background and competitors for a new breaking through Cellular Processing Algorithm. Experimental data model was presented in previous research, however, for the first time an instance generator was created to additionally contribute the paper. It will be available online as open access. First computational experiments demonstrated results obtained by all the algorithms. Results were carefully analyzed and widely commented.

Our current work is focused on computational experiment run. It is planed to solve a vast number of instances using all the algorithms. Moreover, the idea is to provide scalability analysis - with a set of experiments for heuristic algorithms. This type of experiment will be performed for much bigger instances of the problem (with a significantly bigger number of shops and products) to compare all heuristic algorithms. Important part of a discussion will be dispersion analysis. In the full version of this position paper a very detailed description of the computational experiment will follow a vast number of exhaustive tests that will be performed. Moreover, it is planed to describe each algorithm in a very detailed way (including algorithm pseudo-code) to enable repetition and evaluation of these algorithms. Furthermore, an instance generator described within this paper will be available online for open usage. An interesting extension can be made when the model (with instance generator) will be enhanced to complete to analysis several kinds of products and compare the different scenarios.
To alleviate the problem, and also deal with scalability we consider investigating a parallel version of the algorithm on a GPU infrastructure. Moreover, very interesting topic could be to combine a method of selecting column widths for a given set of advertisement on a web page (Marszałkowski and Drozdowski, 2013) with ISOP when preparing online working application.

ACKNOWLEDGEMENTS

This study was partially supported by the FNR (Luxembourg) and NCBiR (Poland), through IShOP project, INTER/POLLUX/13/6466384.

REFERENCES

Blazewicz, J., Bouvry, P., Kovalyov, M., and Musial, J. (2014). Internet shopping with price sensitive discounts. *4OR-Q J Oper Res*, 12(1):35–48.

Blazewicz, J., Kovalyov, M., Musial, J., Urbanski, A., and Wojciechowski, A. (2010). Internet shopping optimization problem. *Int. J. Appl. Math. Comput. Sci.*, 20(2):385–390.

Błazewicz, J. and Musiał, J. (2011). E-Commerce Evaluation–Multi-Item Internet Shopping. Optimization and Heuristic Algorithms. In Hu, B., Morasch, K., Pickl, S., and Siegle, M., editors, *Operations Research Proceedings 2010: Selected Papers of the Annual International Conference of the German Operations Research Society*, pages 149–154. Springer Berlin Heidelberg.

Braun, T., Siegel, H., Beck, N., Bölöni, L., Maheswaran, M., Reuther, A., Robertson, J., Theys, M., Yao, B., Hensgen, D., et al. (2001). A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed computing*, 61(6):810–837.

Clay, K., Krishnan, R., and Wolff, E. (2001). Prices and price dispersion on the web: Evidence from the online book industry. NBER Working Papers 8271, National Bureau of Economic Research, Inc.

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition.

Eiselt, H. and Sandblom, C. (2004). *Decision analysis, location models, and scheduling problems*. Springer-Verlag, Berlin-Heidelberg-New York.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, Freeman.

Ibarra, O. H. and Kim, C. E. (1977). Heuristic algorithms for scheduling independent tasks on nonidentical processors. *J. ACM*, 24:280–289.

Iyigun, C. and Ben-Israel, A. (2010). A generalized weiszfeld method for the multi-facility location problem. *Oper. Res. Lett.*, 38(3):207–214.

Krarup, J., Pisinger, D., and Plastriab, F. (2002). Discrete location problems with push-pull objectives. *Discrete Applied Mathematics*, 123:363–378.

Land, A. and Doig, A. (1960). An automatic method for solving discrete programming problems. *Econometrica,*, 28:497–520.

Lee, H.-G. (1998). Do electronic marketplaces lower the price of goods? *Commun. ACM*, 41(1):73–80.

Luxembourg-Embassy-Copenhagen (2005). E-commerce luxembourg.

Marszałkowski, J. and Drozdowski, M. (2013). Optimization of column width in website layout for advertisement fit. *European Journal of Operational Research*, 226(3):592–601.

Melo, M., Nickel, S., and Saldanha-da Gama, F. (2009). Facility location and supply chain management - a review. *European Journal of Operational Research*, 196(2):401–412.

Musial, J. (2012). *Applications of Combinatorial Optimization for Online Shopping*. NAKOM, Poznan, Poland, 1st edition.

Revelle, C., Eiselt, H., and Daskin, M. (2008). A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3):817–848.

Terán-Villanueva, J. D., Huacuja, H. J. F., Valadez, J. M. C., Pazos Rangel, R. A., Soberanes, H. J. P., and Flores, J. A. M. (2013). Cellular processing algorithms. In Melin, P. and Castillo, O., editors, *Soft Computing Applications in Optimization, Control, and Recognition*, volume 294 of *Studies in Fuzziness and Soft Computing*, pages 53–74. Springer Berlin Heidelberg.

Timmers, P. (1998). Business Models for Electronic Markets. *Electronic Markets*, (2):3–8.

Wojciechowski, A. and Musial, J. (2009). A customer assistance system: Optimizing basket cost. *Foundations of Computing and Decision Sciences*, 34(1):59–69.

Wojciechowski, A. and Musial, J. (2010). Towards optimal multi-item shopping basket management: Heuristic approach. In Meersman, R., Dillon, T., and Herrero, P., editors, *On the Move to Meaningful Internet Systems: OTM 2010 Workshops*, volume 6428 of *Lecture Notes in Computer Science*, pages 349–357. Springer Berlin / Heidelberg.

Wu, M., Shu, W., and Zhang, H. (2000). Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems. In *Heterogeneous Computing Workshop, 2000.(HCW 2000) Proceedings. 9th*, pages 375–385. IEEE.