Paper 46: A Novel Method for Selecting and Materializing Views based on OLAP Signatures and GRASP

Andresson da Silva Firmino[1], Rodrigo Costa Mateus[1], Valéria Cesário Times[1], Lucidio Formiga Cabral[2], Thiago Luís Lopes Siqueira[3], Ricardo Rodrigues Ciferri[4], Cristina Dutra de Aguiar Ciferri[5]

[1]Informatics Center - Federal University of Pernambuco, 50733-970, Recife-PE, Brazil

[2] Informatics Department, Federal University of Paraíba, 58035-000 , João Pessoa, PB, Brazil

[3] São Paulo Federal Institute of Education, Science and Technology, 13565-905, São Carlos, SP, Brazil

[4] Computer Science Department, Federal University of São Carlos, 13565-905, São Carlos -SP, Brazil

[5] Computer Science Department, University of São Paulo at São Carlos, 13560-970, São Carlos-SP, Brazil

{asf2, rcm3, vct}@cin.ufpe.br, prof.thiago@ifsp.edu.br, ricardo@dc.ufscar.br, cdac@icmc.usp.br

Implemented Revisions

We are grateful for the valuable suggestions, which have contributed toward improving the quality of the paper. In the current version, we have revised both the contents and structure of the paper in order to clarify the main contributions of the work done, add explanations to connected parts of the paper, include further details about the conceptual model schema that is proposed in the paper for recording OLAP signatures, add text sentences to better describe each algorithm and give examples of their applications. We have also accepted the suggestions of including a comparison between our GRASP based algorithm and a very recent approach to better assess the impact of our VSP method. The following table contains the reviewers' suggestions and respective solutions implemented.

Reviewer 1	Implemented Solution
2) THEORETICAL FOUNDATION. (i) Lack of explanation about the relationship among three topics: OLAP, view selection problem and GRASP metaheuristic.	**Done**. As requested, we have rewritten the beginning of Section 2 to indicate the relationship among VSP, OLAP signatures and metaheuristics.
2) THEORETICAL FOUNDATION. (ii) Here, the views selection problem focuses three aspects: storage space, processing time for materializing views, and update view. But, the paper just focuses time processing. Why have three aspects been presented here?	**Done**. In our paper, we focus on the views selection problem by proposing a novel method for selecting and materializing views based on OLAP signatures and GRASP. We use storage space requirements to identify which views should be materialized. In fact, we do not focus on the processing time for materializing views and update views. These aspects are not described in Section 2. With regard to the update view aspect, we have included this aspect as a future work.
2) THEORETICAL FOUNDATION. (iii) Theoretical foundation is small and lack of relationship among three topics, and also paper proposal.	**Done**. We have rewritten the beginning of Section 2 to indicate the relationship among VSP, OLAP signatures and metaheuristics, and to explain the association between these three topics and our work. Also, Section 2 was extended with a description of another metaheuristic that was compared to our VSP algorithm.
3) CONCEPTUAL METAMODEL. Some aspects are unclear in the paper: (i) the use of the model.	**Done**. In the paper, we explain the use and the motivation of the conceptual model schema in different sections, such as: - In Section 2, we highlight that "*Another contribution is a conceptual model schema for storing OLAP queries characteristics, referred to as OLAP signatures, which describes OLAP queries submitted by users and helps in identifying significant elements of data cubes used in the analytic processing*." - In Section 3, we detail in the first paragraph the purpose of the proposed conceptual model schema. - In Sections 4.1 and 4.2, we indicate that the conceptual model schema is used for extracting multidimensional data and recording OLAP signatures, respectively.
3) CONCEPTUAL METAMODEL. Some aspects are unclear in the paper: (ii) costs about: storage space, time processing and cost/time	**Done**. In our conceptual model schema given in Section 3, we explained that we store information about sizes in bytes, since storage

Reviewer 1	Implemented Solution
data update.	costs are given by multiplying the number of lines by the size of these lines. This computation indicates the size of a view and is one of the aspects used to select views to be materialized according to the storage space available. Regarding processing time, it is modelled since this is considered as being inversely proportional to the storage costs of views. With regard to the costs of updates, they were included as future work (Section 7) because in the current version of our work, benefits are computed based only on the size and frequency information of views.
3) CONCEPTUAL METAMODEL. Some aspects are unclear in the paper: (iii) Due to the importance of the metamodel, it should be better explained in the text. What has been the base of the model? How could the model be used? Have you done some empirical evaluation?	**Done**. In Section 3, we explained that the main entities of the proposed conceptual model schema were derived from the most significant elements of a data cube and indicated how it is be used to generate OLAP signatures. Further details on the conceptual model schema were given in Section 3 as well. However, the empirical evaluation was seen as out of the scope of this paper and thus, it was included as future work in Section7.
3) CONCEPTUAL METAMODEL. Some aspects are unclear in the paper: (iv) I also suggest better explaining the relationship among model entities. After reading section 4, I understand that measure, cube, level and vertex are used by extractors and query and signature are used by recorder user information. But, in this point of paper this relationship is not clear for reader.	**Done**. The conceptual model schema given in Section 3 was further explained by indicating how the information modelled is related to the algorithms detailed in Section 4. This was made as follows: - In Section 3, we state that "*Therefore, information on measures, cube, levels and vertices are loaded by multidimensional data extractor algorithms discussed in Section 4.1, while query and signature are used to record historical information on the profiles of OLAP users as outlined in Section 4.2.*" - In Section 3, we indicate that "*The frequency information of views is also used to select views to be materialized as further explained in Section 4.2 and is incremented when an instance of the relationship between the query signature and the view with the smallest size is identified*."
4) PROPOSED METHOD. The algorithm	**Done.** In the paper, we explained the main idea

Reviewer 1	Implemented Solution
explanations are some confusing. I suggest the following steps to describe each algorithm: (i) to explain the main idea of algorithm; (ii) to present algorithm and to explain each line of the algorithm, (iii) to show some example about the use, and (iv) lack of computacional analyses (even superficially described).	of our algorithms at the beginning of Section 4 as follows:" *In this section, we introduce the proposed method for selecting and materializing views, which is composed of four phases having the following goals: (i) extract data and metadada from a DW, from its corresponding star schema, and from a data cube schema, and then, compute all possible vertices for this cube (Section 4.1); (ii) record information about the user OLAP queries submitted so far and build the corresponding OLAP signatures (Section 4.2); (iii) select the more beneficial vertices (i.e. views) for the execution of user OLAP queries (Section 4.3); and (iv) materialize the chosen vertices (Section 4.4).*" Also, in the paper, each line of the proposed algorithms is explained in Sections 4.1, 4.2, 4.3 and 4.4. However, due to space limitations, in the current version of our paper, we included complexity analysis as future work (Section 7), and used the experimental setup of Section 5 to exemplify our algorithms and to connect our test results with the algorithms proposed in Section 4. This was made as follows: - In Section 5.1, we state that "*Data generation produced 6 million tuples in the fact table, and the algorithms 1 and 2 listed in Section 4.1 have yielded 14,175 possible materialized views, whose estimated total size of materialization is 6,69 TB."* - In Section 5.3, we highlight that "*Experiments considered the following values of the space available for materialization (in GB): {0,25; 0,5; 1; 2; 3; 4} that represent the following percentages {5; 10; 20; 40; 60; 80}, respectively, on the total space of materialization of all views (i.e. 32 views occupying 5GB in total that were generated by algorithms 3, 4 and 5 of Section 4.3) given as input to GRASP and ACO as well.*"
5) EXPERIMENTS. Why was GRASP compared with PBS? Why is PBS baseline?	**Done.** The reason for using the PBS algorithm in our study is two-fold, as explained in Section 5.3. Firstly, it is a fast algorithm and has broadly been used in recent comparative analysis,

Reviewer 1	Implemented Solution
	providing good results [Kalnis et al. 2002; Zhou et al. 2009; Khan and Aziz 2010]. Secondly, it has a high cardinality of selection because it uses a single criterion of selection, while our GRASP proposal is more complex, is expected to have a greater runtime of selection of views and a low cardinality of selection. Thus, PBS satisfied our requirements for comparison as we aimed at investigating whether the selection of much less views would impair queries runtime. However, this time, the text also includes a comparison between our GRASP based algorithm and a very recent VSP approach, namely Ant Colony Optimization (ACO) [Song, X. and Gao, L. 2010] that selected a small number of views. Please, see Section 5.3.
6) CONCLUSION. Is data structure referring to metamodel?	**Done**. We apologize for this mistake. We replaced the term "data structure" by "conceptual model schema" in Section 7.

Reviewer 2	Implemented Solution
The evaluation does not consider the costs of updates, and the comparison is made against a very old approach (1998).	**Done**. The main contribution of our work is a novel VSP algorithm for selecting and materializing views based on GRASP. A secondary contribution is a conceptual data schema for storing OLAP queries characteristics. This schema helps in describing queries submitted by users and identifies significant elements of data cubes used in the analyses. In the current version, we have validated our ideas by comparing our VSP algorithm to two approaches: PBS (Pick by Size) [Shukla et al. 1998] and ACO (Ant Colony Optimization) [Song, X. and Gao, L. 2010]. Therefore, further evaluation on costs of updates is out of the scope of this paper.
The paper is very well written overall, having just a few typos here and there--please run a spell checker.	**Done**. This time, the text was revised by a native speaker.
One criticism to the organization of the material is that too much space is given to presenting the problem and the solution, and none of which is technically hard nor novel. The most interesting parts of the paper are the experiments, and in more than one occasion the authors mention they run out of space and refer the reader elsewhere for details.	**Done.** Although the current version of our paper includes extra test results as our GRASP based algorithm was compared to another VSP approach namely Ant Colony Optimization (ACO) [Song, X. and Gao, L. 2010], the preliminary sections of our paper were extended because of the following reasons: (1) further details on our conceptual model schema were added to Section 3, as requested by other reviewers; (2) to the best of our knowledge, the VSP solution based on GRASP and OLAP signatures that are discussed in Sections 2 and 3 are novel.
It is hard to assess the impact of the method. This happens for two reasons: (1) not considering updates in the query mix. Many papers in the area, including some cited by the present paper, also report results with updates to the base tables. In fact, even the paper mentions updates as part of the formulation of the problem (e.g., Sec 2.2). But the experiments ignore updates altogether. In this way, one can interpret the results given as a comparison between a greedy solution which is agnostic to the workload (PBS) versus a solution that is based on searching a space which includes information about the workload (GRASP).	**Done**. To ensure the text consistency, any aspect related to the costs of updates was removed from the current version of the paper. Also, this time, we compared our VSP method to another optimization algorithm, called ACO. Similar to GRASP, ACO also includes information about the workload and differently from the PBS, ACO has selected a small number of views to be materialized and its results were compared to solutions found by GRASP. Finally, it is important to note that no parameter settings for PBS, was needed because it lacks input parameters.

Reviewer 2	Implemented Solution
It is not surprising then that the search approach performs better--even less so in light of the parameter tuning performed by the authors (Sec 5.2). The question remaining is then whether the 20% improvement in time and 97% reduction in space are indeed significative, or whether any decent method with proper tuning would result in comparable improvements.	
(2) comparing against an old approach. The main question here is why comparing against PBS alone? The paper mentions other kinds of heuristics in section 6, but does not compare the GRASP approach against any of them.	**Done**. The reason for using the PBS algorithm in our study is two-fold, as explained in Section 5.3. Firstly, it is a fast algorithm and has broadly been used in recent comparative analysis, providing good results [Kalnis et al. 2002; Zhou et al. 2009; Khan and Aziz 2010]. Secondly, it has a high cardinality of selection because it uses a single criterion of selection, while our GRASP proposal is more complex, is expected to have a greater runtime of selection of views and a low cardinality of selection. Thus, PBS satisfied our requirements for comparison as we aimed at investigating whether the selection of much less views would impair queries runtime. However, this time, the text also includes a comparison between our GRASP based algorithm and a very recent VSP approach, namely Ant Colony Optimization (ACO) [Song, X. And Gao, L. 2010]. Please, see Section 5.3.

Reviewer 3	Implemented Solution
There are open issues that need to be clarified as follows: The proposed method generates less materialized views than the PBS approach (97% of reduction). Figure 2(a) shows that the number of selected views using GRASP algorithm is not correlated to the percentage of available storage space. It is not clear why this result is considered a better result?	**Done**. This time, we defined how the amount of space available for materialization (in GB) was computed. This was made as follows: - In Section 5.3., we highlight that*: "Experiments considered the following values of the space available for materialization (in GB): {0,25; 0,5; 1; 2; 3; 4} that represent the following percentages {5; 10; 20; 40; 60; 80}, respectively, on the total space of materialization of all views (i.e. 32 views occupying 5GB in total that were generated by algorithms 3, 4 and 5 of Section 4.3) given as input to GRASP and ACO as well.*" Regarding 97% of reduction, we acknowledge that there was a misunderstanding related to their benefits and therefore, such statement was removed from the current version of our paper.
In page 12, the last paragraph discusses the performance to process OLAP queries using PBS and GRASP materialized views. The authors concludes (page 13, first paragraph) that executing queries using views created by GRASP is more efficient than executing queries using views selected by PBS, considering all query profiles. How can you prove this? It is not clear that this is true for all possible selected sets of materialized views generated by PBS and GRASP. In fact, we can say that the use of SSB workload is good for comparing GRASP and PBS approaches but this is not a proof that GRASP will be better than PBS in all situations.	**Done**. Yes, this was a mistake. We apologize for this. To correct it, the beginning of the fifth paragraph of Section 5.3 was replaced by the following statement: "*Figure 3(a) shows the results, which indicated that executing queries using views created by GRASP **was** more efficient than executing queries using views selected by PBS, considering all query profiles*."
Why the query profile proposed approach is adequate for this kind of experiment. This approach is based on other approach or the authors created it?	**Done**. Yes, we designed the query profiles to help in the execution of our experiments. The purpose of using them is stated in the paper as follows. - In Section 5.2., we indicate that*: "Therefore, the purpose of using query profiles is to evaluate the response time in different scenarios of queries submissions, so that a set of queries may have frequencies of submission that are greater or less than another set of queries."*
In figure 1, the term conceptual metamodel is	**Done**. Yes, it resulted from a misunderstanding.

Reviewer 3	Implemented Solution
misleading. A conceptual model is used for generating data schemas. In this paper this is not the case. This is simply a schema for storing information about OLAP signatures;	We apologize for this mistake. The wrong term was removed from the text. Please see Section 3.
Page 4, 4th paragraph, first line, for a give data- for a given data;	**Done**.
In section 4.1, the explanation of algorithms 1 and 2 are difficult to read. I would suggest to give an intuitive idea of the algorithms and not try to describe them literally;	**Done**. In the paper, we explained the main idea of our algorithms at the beginning of Section 4 as follows:" *In this section, we introduce the proposed method for selecting and materializing views, which is composed of four phases having the following goals: (i) extract data and metadada from a DW, from its corresponding star schema, and from a data cube schema, and then, compute all possible vertices for this cube (Section 4.1); (ii) record information about the user OLAP queries submitted so far and build the corresponding OLAP signatures (Section 4.2); (iii) select the more beneficial vertices (i.e. views) for the execution of user OLAP queries (Section 4.3); and (iv) materialize the chosen vertices (Section 4.4).*" However, we still describe each line of the proposed algorithms, as this was requested by another reviewer.