Traditional database queries follow a simple model: they define constraints that each tuple in the result must satisfy. This model is computationally efficient, as the database system can evaluate the query conditions on each tuple individually.

However, many practical,real-world problems require a collection of result tuples to satisfyconstraints collectively, rather than individually.

we present package queries, a new query model that extends traditional database queries to handle complex constraints and preferencesover answer sets.

We develop a full-fledged package query system,implemented on top of a traditional database engine. Our workmakes several contributions.

First, we design PaQL, a SQL-based query language that supports the declarative specification of package queries.

We prove that PaQL is at least as expressive as integer linear programming, and therefore, evaluation of package queries is in general NP-hard. Second, we present a fundamental evaluation strategy that combines the capabilities of databases and constraint optimization solvers to derive solutions to package queries.

The core of our approach is a set of translation rules that transform a package query to an integer linear program. Third, we introduce an offline data partitioning strategy allowing query evaluation to scale to large data sizes. Fourth, we introduce SKETCHREFINE, a scalable algorithm for package evaluation, with strong approximation guarantees ((1\_e)6-factor approximation).

Finally, we present extensive experiments over real-world and benchmark data. The results demonstrate that SKETCHREFINE is effective at deriving high-quality package results, and achieves runtime performance that is an order of magnitude faster than directly using ILP solvers over large datasets.

Traditional database queries follow a simple model: they define constraints, in the form of selection predicates, that each tuple in the result must satisfy. This model is computationally efficient, as the

database system can evaluate each tuple individually to determine whether it satisfies the query conditions.

However, many practical, real-world problems require a collection of result tuples to satisfy

constraints collectively, rather than individually

A dietitian needs to design a daily meal plan for a patient. She wants a set of three gluten-free meals, between 2,000 and 2,500 calories in total, and with a low total intake of saturated fats.

An astrophysicist is looking for rectangular regions of the night sky that may potentially contain previously unseen quasars.

Regions are explored if their overall redshift is within some specified parameters, and ranked according to their likelihood of containing a quasar Given a task *T*, a pool of individuals *X* with di®erent skills, and a social network *G* that captures the compatibility among these individuals, we study the problem of ¯nding *X0*, a sub-set of *X*, to perform the task. We call this the Team For-mation problem.

We require that members of *X0* not only meet the skill requirements of the task, but can also work ef- fectively together as a team.

We measure e®ectiveness using the *communication cost* incurred by the subgraph in *G* that only involves *X0*. We study two variants of the problem for two di®erent communication-cost functions, and show that both variants are NP-hard.

We explore their connections with existing combinatorial problems and give novel algo-rithms for their solution. To the best of our knowledge, this is the ¯rst work to consider the Team Formation problem in the presence of a social network of individuals.

Experi- ments on the DBLP dataset show that our framework workswell in practice and gives useful and intuitive results.

The success of a project depends not only on the exper-tise of the people who are involved, but also on how e®ec-tively they collaborate, communicate and work together as Part of this work was done when the author was at IBMAlmaden Research Center.

The existence of a social network between individuals isquite common in real scenarios.

In a company, the network may capture the hierarchical organization of the employ-ees. In this case, the graph encodes the fact that people inthe same group or department can communicate easier thanpeople working in di®erent divisions.

In a research commu-nity, the network captures previous successful collaborationsamong scientists. Other examples of social networks be-

tween professionals include LinkedIn

We study the problem of making recommendations when the objects to be recommended must also satisfy constraints or requirements.

In particular, we focus on course recommendations: the courses takenby a student must satisfy requirements (e.g., take 2 out of a set of 5 math courses) in order for the studentto graduate. Our work is done in the context of the CourseRank system, used by students to plan their academic program at Stanford University.

Our goal is to recommend to these students courses that not only help satisfy constraints, but that are also desirable (e.g., popular or taken by similar students).

We develop increasingly expressive models for course requirements, and present a variety of schemes for both checkingif the requirements are satisfied, and for making recommendations that take into account the requirements.

We show that some types of requirements are inherently expensive to check, and we present exact as well as heuristic techniques for those cases.

Although our work is specific to course requirements, it provides insights into the design of recommendation systems in the presence of complex constraints found in other applications.

In traditional applications for recommender systems, the objects that are recommended do not need to satisfy constraints or requirements. For example, customers are not obliged to read certain sets of books, and the recommended books do not need to satisfy constraints.

Thus, traditional recommender systems focus on coming up with a score for each target (e.g., book ormovie) independently, based on indicators such as popularity, target objects read or bought by similar people, and so on. Each target is recommended independently of other targets, and there is no need to check if the recommendedtargets satisfy constraints or requirements.

However, there are important applications where target objects do have constraintsor requirements.

For example, some medical treatments cannot be given concurrently.Or a computer can be configured with up to two optical storage units from a set ofchoices (CD writer, DVD reader, BluRay player, etc.). Or a holiday travel plan must take into account the total budget.

Thus, when we make recommendations we would like to take into account such constraints, in addition to the traditional indicators.Recommendation of an *individual* item to satisfy constraints is not a particularly hard problem: this boils down to an extra pre-recommendation filtering step to prune out those items that do not satisfy constraints. For example, Yelp ([www.yelp.com](http://www.yelp.com)), which offers restaurant recommendations, offers users the option to filter recommendations based on cuisine, distance from a given point, how expensive it is, and so on.

The harder problem, instead, is that of *set recommendations*, recommending a set of items that satisfy several constraints. Apart from travel package recommendations, which are done by some travel websites such as Expedia (www.expedia.com), very few existing real-world systems integrate traditional scoring strategies with set recommendations and hard requirements or constraints.

In this paper we study set recommendations under constraints or requirements.We

do our work in the context of the CourseRank project. As we will discuss in the next

section, CourseRank is a system developed at Stanford for evaluating courses and

planning academic programs, and is in use at over 170 universities today. Working on

a particular application (course recommendations) grounds our research, and lets usprovide solutions for real constraints and real students. Furthermore, as we will see,course requirements are especially rich, so we can explore different types of requirements

and solutions. As a matter of fact, one of the main challenges we address in our

paper is the modeling of complex academic requirements, a problem that as far as we know has not been formally studied.

The area we focus on, academic software, is in itself a multi-billion dollar business

and a major application area for information management systems. Blackboard, just one of the many companies in the domain of education software, has a market cap of 940 million dollars (March 14, 2009). There are over 6000 universities in the USA alone, with over 15M college students, most of whom use software to track courses that they take. Several companies, including Red Lantern (DARS), Jenzabar, Datatel,Sungard (SCT Banner & CAPP), Conclusive Systems, and PeopleSoft (Oracle) have products for course planning (they perform requirement checking but not recommendations).

In spite of the importance of this area, we could not find scientific literature

on course and requirements tracking and recommendations.

Given our experience with CourseRank, and our own experience as students or academic advisors, we see a strong need for recommendations that take into account requirements. For instance, the deployed CourseRank currently offers a primitive requirements check for the five most popular majors (including Computer Science) at Stanford. In the January 1 to March 13 (2009) period, 33% of the Computer Science students who logged onto CourseRank checked the requirements page at least once. This activity indicates a significant interest by students in checking what portion of their requirements have been met. CourseRank already offers a separate course recommendation service which is also popular with students, but the recommendations do not take into account requirements. It is clear to all the academic advisors we work with that the system would be more useful if we recommend not just “interesting” courses, but “interesting” courses that help students graduate! The academic requirements at most universities are complex enough that students often have a hard time identifying good ways to complete their course requirements (and often just forget to explicitly check the requirements).

a new hybrid self-adaptive harmony search combined with a stochastic local search algorithm (SAHS-SLS) to solve the 0-1 multidimensional knapsack problem (MKP). The proposed SAHS-SLS uses SAHS to create harmonies that will be improved with SLS. We propose a dynamic adjustment of the walk probability (*wp*) in SLS and a technique to compute the bandwidth (bw) and the pitch adjusting rate (PAR) in SAHS. The overall method SAHS-SLS is implemented and evaluated on benchmarks in order to measure its performance in solving the MKP. It is compared to other approaches to show its effectiveness. The numerical results are encouraging and demonstrate the benefit of the proposed approach.

**Biographical notes:** Abdellah Rezoug is a PhD student at the University of USTHB. He received his Magister degree in Computer Science from the University of Boumerdes. He is a Lecturer at the Department of Computer Science, University of Boumerdes. His research interests include issues related to the heuristic approaches, optimisation problems, the winner determination problems and the multidimensional knapsack problem. Dalila Boughaci is a Full Professor of Computer Science at the University of USTHB. She received her PhD from the University of Provence (Marseille-France) in 2008 and her Habilitation Post-doctoral Diploma from USTHB University in 2009. Her current research interests are in the areas of evolutionary computation, artificial intelligence and meta-heuristics. She has published several papers on these research topics in journals and conferences and directed several PhD, MS and BS students’ projects. Hei is the Head of the research team: Optimisation, Reasoning and Application of the LRIA Laboratory.

Existing approaches for solving the MKP can be classified into exact and approximate methods. Among the exact approaches proposed for the MKP, we cite: branch and bound (Vimont et al., 2008; Fukunaga, 2011), relaxation (Freville and Plateau, 1997) and dynamic programming (Volgenant and Zoon, 1990). These

approaches have the advantage of efficiency to solve MKP problems of small size and provide exact results. However, the execution time increases exponentially with the size of problem. The approximate approaches can provide good results within reasonable time. Several approaches have been proposed to the MKP such as: tabu search (Aboudi and Jornsten, 1994), simulated annealing (Cho and Kim,1997), genetic algorithm (Khuri et al., 1994; Chu and Beasley, 1998; Yoon et al., 2005; Lin, 2008), ant-colony (Shi, 2006), harmony search (HS) (Zou et al., 2011), evolutionary algorithm (Liu and Liu, 2009), particle swarm optimisation (Li and Li, 2009) as well as others (Egeblad and Pisinger, 2009; Boussier et al., 2010; Hill et al., 2012).

That is always advantageous to combine an evolutionary-based method with a local search to ensure a balance between the global exploration and the local

exploitation of the search space. Motivated by this idea, we propose a hybrid self-adaptive harmony search (SAHS) combined with a stochastic local search (SLS) to solve MKP. SAHS is used to ensure exploration while SLS performs exploitation. First, we improved the SAHS by adding a tuning strategy for the pitch adjusting rate (PAR) and the Bandwidth (*bw*). Then, we apply the local search SLS on every generated solution with a specified Probability (ESP) strategy. The proposed self-adaptive harmony search-stochastic local search (SAHS-SLS) algorithm is evaluated on the well-known MKP benchmarks proposed by Chu and Beasley (1998). The rest of the paper is organised as follows. Section 2 gives a background on the solution representation, the encoding technique, the HS, SLS and the SAHS

approaches. Section 3 details the proposed new approach SAHS-SLS. Section 4 presents the experimentation and gives some numerical results. In Section 5 we discuss the obtained results.

Vacation planning is one of the frequent—but nonetheless laborious—tasks that people engage themselves with online; requiring skilled interaction with a multitude of resources.

This paper constructs intra-city travel itineraries automatically by tapping a latent source reflecting geo-temporal breadcrumbs left by millions of tourists. For example, the popularm rich media sharing site, Flickr, allows photos to be stamped

by the time of when they were taken and be mapped to Points Of Interests (POIs) by geographical (i.e. latitudelongitude) and semantic (e.g., tags) metadata.Leveraging this information, we construct itineraries following a two-step approach. Given a city, we first extract photo streams of individual users. Each photo stream provides estimates on where the user was, how long he stayed at each place, and what was the transit time between places. In the second step, we aggregate all user photo streams into a POI graph. Itineraries are then automatically constructed from the graph based on the popularity of the POIs and

subject to the user’s time and destination constraints.

We evaluate our approach by constructing itineraries for several major cities and comparing them, through a “crowdsourcing” marketplace (Amazon Mechanical Turk), against itineraries constructed from popular bus tours that are professionally

generated. Our extensive survey-based user studies over about 450 workers on AMT indicate that high quality itineraries can be automatically constructed from Flickr data.

Travel itinerary planning is often a difficult and time consuming task for a traveler visiting a destination for the first time.

It involves substantial research to identify points of interests (POIs) worth visiting, the time worth spending at each point, and the time it will take to get from one place to another. Without any prior knowledge, one must either rely on (1) travel books, (2) personal travel blogs, or (3) a combination of online resources and services such as travel guides, map services, public transportation sites, and human intelligence to piece together an itinerary

.All these options have shortcomings. Travel books do notcover all cities/locations and, perhaps more importantly, arenot free. Personal travel blogs reflect a single person’s view,with no guarantees provided over the writer’s experience orthe amount of preparation invested in planning the trip.Finally, compiling an itinerary by selecting individual POIsand researching their to’s and fro’s is a task which is bothtime consuming and requires significant search expertise.Fortunately, with the advancement of digital photographyand the rapid rise of rich media sharing sites such as Flickr(http://www.flickr.com/), millions of travelers are now sharingtheir travel experiences through rich media data such asphotos.

More interestingly, users are increasingly associatingshared media with rich contextual information.

Flickr photos, for example, are usually time-stamped by the dateand time of when they were taken. Furthermore, they areoften tagged with geographical information (i.e., latitudesand longitudes), which can be easily mapped to the POIs.Even more frequently, the photos are associated with textualmetadata such as tags, titles, notes, and descriptions.Such shared photos can be seen as billions of geo-temporal breadcrumbs that can promisingly serve as a latent source reflecting the trips of millions of users.

Our goal is thereforetoautomatically construct travel itineraries at a large scale from thosebreadcrumbs.

More specifically, by analyzingthese breadcrumbs associated with a person’s photo stream, one can deduce the cities visited by a person, which POIs that person took photos at, how long that person spent at each POI, and what the transit time was between POIs visited in succession.

By aggregating such timed paths of many users, one can construct itineraries that reflect the “wisdom” of touring crowds. Each such itinerary is comprised of a sequence of POIs, with recommended visit times and approximate transit times between them.

The tasks described above raise several key challenges that are tag-related, geo-related, or time-related.

Tags are used to capture different user intents. For example, a photo of a person named Paris taken in NYC may be tagged by theperson’s name. City and POI names have different variants.

For example, “NYC”, “Manhattan” and “The Big Apple” all relate to NYC. Geo-location information can be misleading.

For example, pictures of a landmark can be taken from afar such as a picture of the Brooklyn Bridge taken from atop the Empire State Building in NYC. In this case the latitude and longitude information may not match the pictured landmark. As for time-related challenges, some travelers try to maximize the number of POIs they visit, while others like to leisurely tour fewer POIs. Backpackers move between places faster than a family of four can.

In summary, the association of photos to cities/POIs needs to be assessed carefully, and the construction of travel itineraries from photos must address all the challenges raised above.

In addressing these challenges and others, we make the following contributions:

1. We introduce a novel end-to-end approach that starts with the analysis of latent information reflected in social media sharing sites, and ends with the synthesis of practical information in the form of travel itineraries.

2. As an initial implementation of our approach, we apply a pipeline of multiple heuristics that together extract reliable granular evidence of individual tourists’ trips to a destination from Flickr photos.

3. We aggregate the individual trips to form a graph representing collective touristic behavior, and adapt a solution of the Orienteering problem to efficiently generate intra-city travel itineraries from the graph.

An extensive survey based user study eliciting feedback from 450 users on Amazon’s Mechanical Turk platform validatedour system’s ability to generate high quality travel itineraries for popular touristic cities. The rest of the paper is organized as follows:

surveys related work. Section 3 presents some basic terminology and details how we process Flickr photos to derive timed travel paths of many users. Then, in

we aggregate those paths and generate itineraries. Section 5 reports

on our user study, conducted on Amazon Mechanical Turk. Section 6 presents future challenges and conclusions.

Three-staged patterns are often used to solve the 2D cutting stock problem of

rectangular items. They can be divided into items in three stages: Vertical cuts divide theplate into segments;

then horizontal cuts divide the segments into strips, and finally verticalcuts divide the strips into items.

An algorithm for unconstrained three-staged patterns is presented, where a set of rectangular item types are packed into the plate so as to maximize the pattern value, and there is no constraint on the frequencies of each item type.

It can be used jointly with the linear programming approach to solve the cutting stock problem.

The algorithm solves three large knapsack problems to obtain the optimal pattern: One for the item layout on the widest strip, one for the strip layout on the longest segment, and the third for the segment layout on the plate. The computational results indicate that the algorithm is efficient.

Cutting problems appear in many industrial areas, such as the cutting of metal plate, platen glass and wood panel into rectangular items. Good algorithms for generating cutting patterns are useful for better material utilization..

A three-staged pattern can be divided into items in three stages. It is an X-pattern if the first stage cuts are vertical; it is a Y-pattern otherwise. Figure 1 shows the cutting process of a three-staged X-pattern, where the numbers denote the item types and the arrows denote the cuts. Vertical cuts divide the plate into three segments at the first stage (Fig. 1a). Each segment consists of only horizontal strips. Although by intuition the first segment contains one vertical strip, it can be seen as consisting of nine horizontal strips, each of which contains only one piece of type 1. Horizontal cuts divide the segments into strips at the second stage , and vertical cuts divide the strips into items at the third stage .

shows a three-staged Y-pattern. It contains three segments arranged vertically from bottom to top. Each segment can be seen as consisting of only vertical strips.

In this paper we propose a new hybrid heuristic approach that combines the Quantum Particle Swarm Optimization technique with a local search method to solve the Multidimensional Knapsack Problem. The approach also incorporates a heuristic repair operator that uses problem-specific knowledge instead of the penalty function technique commonly used for constrained problems. Experimental results obtained on a wide set of benchmark problems clearly demonstrate the competitiveness of the proposed method compared to the state-of-the-art heuristic methods.

Multidimensional Knapsack Problem (MKP), which seeks to find a subset of items that maximizes a linear objective function while satisfying a set of linear capacity constraints.

Many practical engineering design problems can be formulated

as the 0–1 MKP, such as, cutting stock (Gilmore and Gomory, 1966), project selection (Petersen, 1967), cargo loading problems (Shih, 1979), capital budgeting (Weingartner, 1966), databases and processor allocation in distributed systems (Gavish et al., 1982) or the daily management of a satellite (Vasquez and Hao, 2001). Given the practical and the theoretical importance of the 0–1 MKP, this

problem has been widely studied and solved by many exact as well as heuristic methods.

The reader is referred to Freville (2004), Puchinger et al. (2010) and Varnamkhasti (2012) for a comprehensive and recent annotated bibliography.

Exact methods include dynamic programming (Gilmore and Gomory, 1966; Green, 1967; Weingartner and Ness, 1967), hybrid dynamic programming methods (Bertsimas and Demir, 2002; Balev et al., 2008; Wilbaut et al., 2006), branch and bound algorithms (Fayard and Plateau, 1982; Gavish and Pirkul, 1985; Vimont et al., 2008; Mansini and Speranza, 2012) and hybrid approaches combining constraint programming and integer linear programming (Oliva et al., 2001; Boussier et al., 2010).

The major drawback of these methods remains the temporal complexity when dealing with large instances.

Therefore, many researchers focus on heuristic and meta-heuristic search methods which can produce solutions of good qualities in a reasonable amount of time. Relevant methods include tabu search (Vasquez and Hao, 2001; Dammeyer and Voss, 1993; Glover and Kochenberger, 1996;

Hanafi andn Freville, 1998; Vasquez and Vimont, 2005),genetic algorithm (Chu and Beasley, 1998; Berberler et al., 2013; Martins et al., 2014), simulated annealing (Leung et al., 2012; Rezoug et al., 2015), ant colony optimization (Parra-Hernandez and Dimopoulos, 2003; Kong et al., 2008; Ke et al., 2010; Fingler et al., 2014), filter-and-fan algorithm (Khemakhem et al., 2012), particle swarm optimization (Kong et al., 2006; Wan andn Nolle, 2009; Chen et al., 2010; Ktari and Chabchoub, 2013; Tisna, 2013; Beheshti et al., 2013; Chih, 2015) and so on.

In this paper, we propose an efficient hybrid heuristic approach to solve the 0–1 MKP that effectively combines a relatively recent evolutionary computation technique, the Quantum Particle Swarm Optimization (QPSO), with a local search method. We propose to use QPSO in combination with a heuristic repair operator utilizing problem-specific knowledge, instead of the penalty function technique

usually used to avoid the violation of problem constraints.

We apply this repair operator to amend infeasible solutions or to improve feasible solutions.

In this way, it ensures that the search process will be always guided through a feasible solution space.

The aim of this work is twofold: (i) To investigate the effectiveness of an improved QPSO algorithmwhen dealing with an NP-hard combinatorial optimization problem such as the 0–1 MKP. (ii) To suggest an efficient hybrid approach that combines QPSO with a local searchmethod in the aim to benefit from the good exploitation (intensification) of the search space offered by a local searchmethod algorithm and the good exploration (diversification) and the fast convergence of the modified QPSO method. Note that the proposed Special attention should be given to the ways the problemspecific information could be applied into some repair operators.

The remainder of this paper is organized as follows. describes the basic features of the classical particle swarm optimization (PSO) technique for continuous optimization and then reviews the fundamental principles of the Binary PSO method (BPSO).

introduces our QPSO algorithm to solve the 0– 1 MKP, whereas describes the specific MKP repair operator. describes the local search to repair infeasible solutions and to improve feasible solutions.

presents and discusses the experimental results obtained over a wide set of benchmark problems. concludes with a summary of major results and suggestions for future researches

Branch-and-Bound search is a basic algorithm for solving combinatorial optimization problems. Here we introduce a new lower-bounding methodology that can be incorporated into any branch-and-bound solver, and demonstraint its use on the MaxSAT constraint optimization problem. The approach is to adapt a minimum-height equivalent transformation framework that was first developed in the context of computer vision.

We present efficient algorithms to realize this framework within the MaxSAT domain, and demonstrate their feasibility by implementing them within the state-of-the-art MAXSATZ solver.

We evaluate the solver on test sets from the 2009 MaxSAT competition;

we observe a basic performance tradeoff whereby the (quadratic) time cost of computing the transformationsn may or may not be worthwhile in exchange for better bounds and more frequent pruning. For specific test sets, the trade-off does result in significant improvement in both prunings and overall run-timen MaxSAT is an optimization problem whose theoretical and practical importance has motivated a growing body of research on exact solvers, e.g. (Li, Many`a, and Planes 2007; Heras, Larrosa, and Oliveras 2008; Lin, Su, and Li 2008;

Ans´otegui, Bonet, and Levy 2009). Solutions are variable assignments that maximize the weight of the clauses that they satisfy–or equivalently, by convention the goal is to minimize the weight of unsatisfied clauses. As a discrete optimization problem, MaxSAT is amenable to branch-andbound search; for this approach to work it is critical to computetight lower bounds on the weight of clauses that must go unsatisfied upon completing a partial assignment, in order to prune the search space below said assignment whenever the lower bound exceeds an upper bound representing the best solution found so far.

Typically, such lower bounds have been produced by applying resolution-like inference rules whenever fixing a variable during search (Li, Many`a, and Planes 2007).

Here we introduce a new framework for constructing MaxSAT lower bounds that can be incorporated into any branch-and-bound solution method. The conceptual basis for this framework is “minimum-height equivalent transformation”(“MHET”), which derives lower bounds by optimistically assuming that we can achieve full problem height, i.e., that we can achieve the maximum score for each clause.

Making this bound non-trivial first requires an extension to the language of MaxSAT problems, where clauses can now give varying weights to different configurations of their variables. Then, given a particular basic problem, we can seek a problem in the extended space of problems that is equivalent in how it scores any variable assignment, but has minimal height. The concept of MHET originates from a formal analysis of vision problems that was originally developed

in the Soviet Union (in the absence of high-powered computers!), and that was recently reviewed in the context of relating probabilistic reasoning to constraint satisfaction (Schlesinger 1976; Werner 2007).

The primary contribution of this paper is to adapt the MHET framework to MaxSAT, introducing representations and algorithms that make the framework tractable for contemporary problems in clausal normal form. The resulting adaptation can be seen as a generalization of existing inference procedures that aspires to produce tighter bounds.

We have also implemented the bounding technique withinthe state-of-the-art MAXSATZ solver, and assessed its usefulness on nineteen test sets representing a broad variety of MaxSAT challenge areas and applications. In some of these

sets, performing equivalent transformations yields an overall improvement in both the number of prunings and the overall runtime. In the remainder, the methodology still increases the number of prunings, but its computational overhead results

in longer overall runtimes.

motivates and defines the notion of minimumheight equivalent transformation for MaxSAT, while 3 presents efficient algorithms for calculating problem height and finding MHET’s to local optimality. Section 4 describes the implementation of MHET and presents empirical results.

Finally, Section 5 relates the framework to existing research and makes concluding observations.

In this paper, we present PACKAGEBUILDER, a system that extends query engines to support package generation. A package is a collection of tuples with certain global properties defined on the collection as a whole. In contrast to traditional query answers, where each answer tuple needs to satisfy the query predicate constraints, each answer package needs to satisfy global constraints on the collection of tuples: e.g., a package of recipes that collectively do not exceed 2,200 calories. PACKAGEBUILDER introduces simple extensions to the SQL language to support package-level predicates, and includes a simple interface that allows users to load datasets and interactively specify package queries. Our system allows users to interactively navigate through the result packages, and to provide feedback by fixing tuples within a package. PACKAGEBUILDER automatically processes this feedback to refine the package queries, and generate new sets of result

Traditional database queries define constraints (selection predicates) that each tuple in the result needs to satisfy. While traditional SQL queries are undoubtedly expressive and powerful, they fall short in scenarios that require a set of answer tuples to satisfy constraints collectively. Such scenarios arise in a variety of applications: Investment portfolio: A broker wants to construct an investment portfolio for one of her clients. The client has a budget of $50K, wants to invest at least 30% of the assets in technology, and wants a balance of short-term and long-term options. The broker cannot select each stock option individually, but rather needs to find a stock package that satisfies all these constraints collectively. Meal planner: An athlete needs to put together a dietary plan in preparation for a race. She wants a high-protein set of three meals for the day, that are between 2000 and 3000 calories in total. All meals should be gluten-free. It is easy to exclude meals that include gluten, as this condition can be checked for each meal (tuple) individually with a regular selection predicate. Other constraints need to be verified collectively over the entire package.

Vacation planner: A couple wants to organize a relaxing vacation at a tropical destination. They do not want to spend more than $2,000 in flights and hotel combined, and they want to be in walking distance from the beach, unless they can fit a rental car in their budget, in which case they are willing to stay farther away. Building the ideal vacation package is challenging, as the choice of hotel affects the choice of other elements in the package (e.g., flights and car rental). In this paper we present PACKAGEBUILDER, a system that augments database functionality to support the creation of packages. A package is a collection of tuples that individually satisfy base constraints and collectively satisfy global constraints. The base constraints are equivalent to regular selection predicates, and can be evaluated individually for each tuple. For example, in the meal planner application, the gluten-free restriction is a base constraint, as it can be verified independently on each meal. In contrast, the requirement that total calories should be within 2,000 to 2,500 calories: it cannot be evaluated on each meal individually, but needs to be assessed over a collection of meals. In this paper we show how to build such packages from database data using package builder queries (PBQs). Our system addresses three main challenges: Language specification: Even though many use cases motivate support for PBQs, this class of queries remains largely unsupported with few tools targeting domain-specific packages (e.g., CourseRank supports building course packages [2]). As part of this work, we will present PaQL, a declarative query language that supports package specifications. PaQL is designed with simple extensions to standard SQL, so those familiar with SQL should find it intuitive and easy to use (Section 3). Interactive specification: Even traditional SQL queries can often be challenging for novice DBMS users to specify. To enable user-friendly database applications, several systems now employ application-independent visual metaphors for SQL query specification [9, 3, 4]. PBQs are fundamentally harder to express and evaluate compared to traditional SQL, therefore, it is increasingly important to provide visual paradigms to guide users through building a query, as well as navigating and possibly refining the results. PACKAGEBUILDER offers an interactive representation of datasets that guides users in specifying base and global constraints for their packages. The system interface also allows users to easily navigate through the solution space by visualizing the result space, and to refine the result packages. (Section 4). Evaluation: In traditional database queries, the size of the answer is polynomial in the size of the input data. This is not true for package queries: If n tuples satisfy the base constraints of a package, there are Ω(2 n ) candidate packages that can satisfy the user’s global constraints. This makes the evaluation of PBQs particularly

challenging. With an exponential search space, efficiently searching for packages that satisfy our users’ constraints requires applying non-trivial pruning techniques and search heuristics (Section 5). We proceed to describe the main three aspects of our system that are motivated by these challenges.

Branch-and-Bound search is a basic algorithm for solving combinatorial optimization problems. Here we introduce a new lower-bounding methodology that can be incorporated into any branch-and-bound solver, and demonstraint its use on the MaxSAT constraint optimization problem. The approach is to adapt a minimum-height equivalent transformation framework that was first developed in the context of computer vision. We present efficient algorithms to realize this framework within the MaxSAT domain, and demonstrate their feasibility by implementing them within the state-of-the-art MAXSATZ solver. We evaluate the solver on test sets from the 2009 MaxSAT competition; we observe a basic performance tradeoff whereby the (quadratic) time cost of computing the transformations may or may not be worthwhile in exchange for better bounds and more frequent pruning. For specific test sets, the trade-off does result in significant improvement in both prunings and overall run-time.

MaxSAT is an optimization problem whose theoretical and practical importance has motivated a growing body of research on exact solvers, e.g. (Li, Manya, and Planes 2007; ` Heras, Larrosa, and Oliveras 2008; Lin, Su, and Li 2008; Ansotegui, Bonet, and Levy 2009). Solutions are variable ´ assignments that maximize the weight of the clauses that they satisfy–or equivalently, by convention the goal is to minimize the weight of unsatisfied clauses. As a discrete optimization problem, MaxSAT is amenable to branch-andbound search; for this approach to work it is critical to compute tight lower bounds on the weight of clauses that must go unsatisfied upon completing a partial assignment, in order to prune the search space below said assignment whenever the lower bound exceeds an upper bound representing the best solution found so far. Typically, such lower bounds have been produced by applying resolution-like inference rules whenever fixing a variable during search (Li, Manya, and ` Planes 2007). Here we introduce a new framework for constructing MaxSAT lower bounds that can be incorporated into any branch-and-bound solution method. The conceptual basis for this framework is “minimum-height equivalent transformation” (“MHET”), which derives lower bounds by optimistically assuming that we can achieve full problem height, i.e., that we can achieve the maximum score for each clause. Making this bound non-trivial first requires an extension to the language of MaxSAT problems, where clauses can now give varying weights to different configurations of their variables. Then, given a particular basic problem, we can seek a problem in the extended space of problems that is equivalent in how it scores any variable assignment, but has minimal height. The concept of MHET originates from a formal analysis of vision problems that was originally developed in the Soviet Union (in the absence of high-powered computers!), and that was recently reviewed in the context of relating probabilistic reasoning to constraint satisfaction (Schlesinger 1976; Werner 2007). The primary contribution of this paper is to adapt the MHET framework to MaxSAT, introducing representations and algorithms that make the framework tractable for contemporary problems in clausal normal form. The resulting adaptation can be seen as a generalization of existing inference procedures that aspires to produce tighter bounds. We have also implemented the bounding technique within the state-of-the-art MAXSATZ solver, and assessed its usefulness on nineteen test sets representing a broad variety of MaxSAT challenge areas and applications. In some of these sets, performing equivalent transformations yields an overall improvement in both the number of prunings and the overall runtime. In the remainder, the methodology still increases the number of prunings, but its computational overhead results in longer overall runtimes. Section 2 motivates and defines the notion of minimumheight equivalent transformation for MaxSAT, while Section 3 presents efficient algorithms for calculating problem height and finding MHET’s to local optimality. Section 4 describes the implementation of MHET and presents empirical results. Finally, Section 5 relates the framework to existing research and makes concluding observations.

The harmony search (HS) method is an emerging meta-heuristic optimization algorithm. However, like most of the evolutionary computation techniques, it sometimes suffers from a rather slow search speed, and fails to find the global optimum in an efficient way. In this article, a hybrid optimization approach is proposed and studied, in which the HS is merged together with the opposition-based learning (OBL). The modified HS, namely HS-OBL, has an improved convergence property. Optimization of 24 typical benchmark functions and an optimal wind generator design case study demonstrate that the HS-OBL can indeed yield a superior optimization performance over the regular HS method.