

On combining numerical optimization techniques with a belief merging approach

Oscar Chávez-Bosquez¹, Pilar Pozos-Parra², and Betania Hernández-Ocaña¹

¹ División Académica de Informática y Sistemas, Universidad Juárez Autónoma de Tabasco {oscar.chavez,betania.hernandez}@ujat.mx

² Instituto Tecnológico Superior de Occidente mariapozos@iteso.mx

Abstract. Metaheuristics are well-known numerical algorithms successfully used to solve different combinatorial optimization problems. On the other hand, belief merging is a logic-based technique used to fusion potentially conflicting pieces of information coming from different sources. Although these two techniques solve different kinds of problems, there are many real-life scenarios requiring both numerical and symbolic approaches. In this paper, we propose a general hybrid framework combining metaheuristics and belief merging. At a very high level, our proposal consists of two black-box modules: an optimization engine and a belief merger. For the first module we propose implementing a Tabu Search, a fast, robust and reliable metaheuristic. For the second module we propose using the Δ_{ps} (PS-Merge) belief merging operator, a versatile operator whose main advantage is that it can operate over inconsistent belief bases; it outperforms other operators in a previous benchmark study. Finally, we describe our proposal and provide insights about its usefulness solving a motivating example.

Keywords: Hybrid framework · Metaheuristics · Knowledge Representation.

1 Introduction

Combinatorial optimization seeks to obtain the “optimal” solution among a set of possibilities. Examples range from popular games like Sudoku, classic problems like the Boolean Satisfiability, to real-life scenarios like the Traveling Salesman Problem. The solution is encoded with discrete variables and typically consist of an integer number, a subset, a permutation, or a graph structure [1].

Definition 1. A *Combinatorial Optimization Problem (COP)* $P = (S, f)$ is defined by:

- A set of variables $X = \{x_1, \dots, x_n\}$.
- Variable domains D_1, \dots, D_n .
- Constraints among variables.
- An objective function f to be maximized (or minimized), where $f : D_1 \times \dots \times D_n \rightarrow R^+$.

- The set of all possible, feasible assignments is $S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ satisfies all the constraints}\}$
- S is the solution space, as each element of the set can be seen as a candidate solution. To solve a COP, we must find a solution $s^* \in S$ with maximum (in the case of maximization) objective function value: $f(s^*) \geq f(s) \forall s \in S$. s^* is the globally optimal solution of (S, f) , and the set $S^* \subseteq S$ is the set of globally optimal solutions.

There are many algorithms to deal with COPs, usually classified into two categories: *complete* or *approximate*. Complete algorithms guarantee to find an optimal solution in a time rate that grows as a polynomial function of the size of the problem. On the other hand, approximate algorithms can obtain acceptable solutions in a significantly reduced amount of time.

Among approximate algorithms, metaheuristics are tools that generate approximate solutions in reasonable times to complex COPs. They include one or more heuristic methods using a higher-level strategy (hence ‘meta’). Heuristics inside of a metaheuristic are considered a black-box as little (if any) prior knowledge is known about it by the metaheuristic, so that it may be replaced with a different heuristic [2].

On the other hand, the integration of information or knowledge from several sources is a relevant problem at present, as many areas need to merge different pieces of data to integrate a common and structured set of information, such as expert opinion fusion in risk analysis, sensor fusion or logic [4]. A number of approaches to tackle these problems have been developed. Among them, Belief Merging is a technique based on model theory and propositional logic that has been successfully applied on complex real-life problems such as IBM’s QRadar exploit detection system [13] and cancer diagnosis [9].

There are hard COPs where a solely numerical approach can be used to solve it, such as in optimization and search. However, some COPs may include the integration of (possibly inconsistent) data, so we propose a high-level model to manage the problem from two points of view: Metaheuristic optimization and Belief merging.

The following core questions determine if our framework is suitable for a given problem:

1. *Is it a Combinatorial Optimization Problem?*
If yes, a metaheuristic may fit the problem.
2. *Does the problem need to combine different and conflicting information pieces?*
If so, belief merging should be used in order to produce a single consistent knowledge base which retains as much information as possible.
3. *Can the problem be split into two sub-problems (numerical and symbolical)?*
If it is the case, this means that two different approaches can be used to solve the problem.

If the third question is affirmative, then we can employ our framework to solve it.

The following include a background on metaheuristics and belief merging formalisms (section 2), the description of our proposal (section 3), a motivating example (section 4) and conclusions (section 5) .

2 Background

2.1 Metaheuristics

Recently, metaheuristics have gained popularity over mathematical programming methods due to their flexibility and robust behavior to solve real-life COPs [14]. Metaheuristics simulate or emulate processes or behavior inspired by nature. A common classification include Evolutionary Algorithms (e.g., Genetic Algorithms, Differential Evolution), Swarm Intelligence (e.g., Ant Colony Optimization, Bacterial Foraging Algorithm) and Global Search Algorithms (e.g., Simulated Annealing, Tabu Search) [12].

The family of Global Search Algorithms is among the most efficient metaheuristics. They manage a local search procedure (neighborhood exploring) in order to combine exploration and exploitation conveniently. Local search algorithms start from some initial solution and iteratively try to replace the current solution with a better one in a defined neighborhood [1].

Definition 2. *A neighborhood is a function $N : S \rightarrow 2^S$ that assigns to every $s \in S$ a set of neighbors $N(s) \subseteq S$. $N(s)$ is called the neighborhood of S .*

Global Search Algorithms use the neighborhood structure along with a specific strategy to escape from local minima.

Definition 3. *A local minimum solution w.r.t. a neighborhood N is a solution s' such that $\forall s \in N(s') : f(s') < f(s)$.*

The term Tabu Search was coined in the same paper that introduced the term metaheuristic [8]. A distinguishing feature of this algorithm is the use of adaptive memory and specific associated problem-solving strategies. Tabu Search provides the origin of the memory-based and strategy-intensive focus in the metaheuristic literature, as opposed to methods that are memory-less. It is also responsible for emphasizing the use of structured designs to exploit historical patterns of search, as opposed to processes that rely almost exclusively on randomization [5].

The particular characteristic of the Tabu Search is a short-term memory called *tabu list* containing moves recently applied. This list is to disallow moves that reverse the effect of recent moves, adding them a sort of forbidden status. However, from time to time a tabu move can reach a better solution space. Thus, *aspiration criteria* are implemented in order to revoke the tabu status of a move. Many criteria have been proposed, but the most widely used aspiration criterion consists in allowing a tabu move if it results in a solution with a better objective value than the current best-known solution. It has been demonstrated in numerous research and real-life projects that Tabu Search finds good approximations to the optimal solution for large combinatorial problems [7].

In general, there are plenty of proposals on metaheuristic hybridization. The most popular approach is a hybrid of Evolutionary Algorithm and a Local Search Algorithm [12]. However, the combination of a metaheuristic with a symbolic approach such as Belief Merging has not been proposed in the literature.

2.2 Belief merging

Belief merging is a knowledge fusion approach for the integration of knowledge coming from different sources. This logic-based technique is defined as the operation of combining information from a set of (possibly conflicting) belief bases obtained from different sources to produce a single and consistent belief base [15].

In this particular case, each belief base corresponds to the knowledge or preferences of any entity of the problem, e.g., stakeholder, sensor, robot, or service. By merging these belief bases, we obtain a belief base that represents the consensus of preferences, or the knowledge of the group.

Belief merging is based on propositional logic, with the significant number of proposals based on model theory. So, in this theory we consider a language $\mathcal{L}(P)$ of propositional logic using a finite ordered set of symbols (atoms or variables) $P = \{p_1, p_2, \dots, p_n\}$. A propositional formula (or a set of formulas) ψ is conformed by a subset of variables from P with the set of logic connectives $\rho = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ in the classical way. $\mathcal{P}(\psi)$ denotes the set of atoms appearing in ψ . $|\mathcal{P}|$ denotes the cardinality of set \mathcal{P} . A literal l is an atom or the negation of an atom. A term D is a finite conjunction of literals i.e., $D = l_1 \wedge \dots \wedge l_k$, with $l_i = p_j$ or $l_i = \neg p_j$. The Disjunctive Normal Form (DNF_ψ) of every formula $\psi \in \mathcal{L}(P)$ is a disjunction of terms $\text{DNF}_\psi = D_1 \vee \dots \vee D_m$, such that $\psi \equiv \text{DNF}_\psi$. An *interpretation* is a function w from P to $\{1, 0\}$ (the classical truth values 1 representing *true* and 0 representing *false*). The set of all possible interpretations is denoted as \mathcal{W} , with elements denoted by vectors of the form $(w(p_1), \dots, w(p_n))$. A model of ψ is an interpretation such that $w(Q) = 1$ once w is extended in the usual way over the connectives, and the set of models of a formula ψ will be denoted by $\text{mod}(\psi)$. ψ is consistent iff there exists a model of ψ .

Apart the classical definitions in model theory, we need the following definitions [11]:

Definition 4. *A belief base K is a finite set of formulas of $\mathcal{L}(P)$ representing the beliefs of a stakeholder. We also identify K with the conjunction of its elements, for example if $K = \{\psi_1, \psi_2, \dots, \psi_n\}$ then $K = \psi_1 \wedge \psi_2 \wedge \dots \wedge \psi_n$.*

Definition 5. *A belief profile $E = \{K_1, \dots, K_m\}$ denotes the beliefs of the group of stakeholders involved in the fusion process. E is a multiset of m belief bases, as two or more stakeholders may have identical beliefs.*

Several belief merging operators have been defined and characterized in a logical way. It is common to represent the result of a merging operator as a set

of interpretations instead of belief bases, such that, if Δ is an operator and E a belief profile, then $mod(\Delta(E))$ represents the set of interpretations resulting from the fusion. However, it is possible to obtain the formula representing the belief base from the set of interpretations, except equivalences.

Belief merging operators are categorized into two main families: model-based operators and formula-based operators. Two subclasses of merging operators are considered [10]:

1. **Majority merging operators** its goal is to satisfy the beliefs of all stakeholders as a whole. If a subset of beliefs in the belief profile is repeated enough times, then it is the one that prevails.
2. **Arbitration merging operators** focus on satisfying as much as possible each stakeholder. This subclass of operators selects the median possible choices.

Δ_{ps} (PS-Merge) is a flexible operator classified as a majority-merging operator but can be refined tending to be an arbitration operator. Unlike cited operators, it can operate over inconsistent belief bases. It is not based on the classic measure distance between models (Hamming distance), but in the notion of *Partial Satisfiability* [16]. This notion allows savings at computing the models of belief bases.

Notably, Δ_{ps} refines the results of other Belief Merging operators in consensus decision-making scenarios [17]. Thus, we propose Δ_{ps} to conform the Belief Merging module of our framework.

3 Hybrid framework

We divide our proposal into two modules, each one performing a task of the problem solution. Each module uses different inputs. As significant data pre-processing may be required, it may be an essential part of the framework. This step may involve the creation of helper data structures, data cleaning or the translation of data formats.

In the first phase, we suggest using a metaheuristic algorithm to solve the optimization part of the problem, we propose specifically the Tabu Search.

The optimization part of the problem should be the first stage of the process, as it produces a preliminary solution which in turn is used by the following module.

For the second phase, a belief merging operator, specifically Δ_{ps} , is used to make the data fusion. At this stage, the input of this stage is the preliminary solution built up in the previous stage. We need to translate this numerical solution into a knowledge base format. This module also needs the other belief bases to merge.

Figure 1 outlines our proposal.

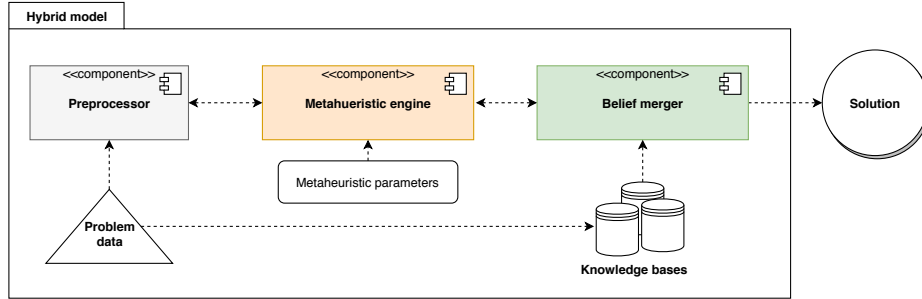


Fig. 1. Hybrid framework high-level view.

3.1 Module 1: Numerical optimization engine

The optimization is performed by this module using Algorithm 1. This Tabu Search implementation is based on the original tutorial by Fred Glover [6].

Main elements include:

Tabu list. It is a queue containing the *disallowed* moves. Its size, the *tabu tenure*, determines the number of iterations that a move is classified as tabu. Tabu tenure may be static or dynamic, depending on the problem type and instance size.

Aspiration criterion. Conveniently called “strategic oblivion,” defines the revocation way of a tabu status over a particular move. In this case, the condition to override tabu status is to accept any move that reaches to a better solution.

Initial solution. An initial solution should be provided in order to take advantage of the deterministic nature of Tabu Search.

Stop criterion. There are many conditions to stop the optimization process. Algorithm 1 ends according to one of the following criteria:

- Fixed number of iterations.
- Limit time, in seconds.
- Find a solution s^* with a given objective function value.
- Number of iterations with no improved solution.

3.2 Module 2: Belief merger

This module has the following components:

Symbolic translator. A belief merging operator computes models of the language, so this component converts the output of Module 1 into a knowledge base format.

Definition 6. Let S_{m_1} be the partial solution generated by Module 1. Then, translating each variable $\varphi \in S_{m_1}$ into a logical variable $\hat{\varphi}$ will give the belief base $S_{m_1}^{\text{logic}} = \{\hat{\varphi}_1, \dots, \hat{\varphi}_n\}$.

Algorithm 1: Tabu Search with simple aspiration criterion.

Input: Initial solution s
Output: Best solution s^*

```

1  $k \leftarrow 1$ ;
2  $s^* \leftarrow s$ ;
3 while not stop criterion is met do
4   Identify  $N(s)$ ; // neighborhood
5   Identify  $T(s, k)$ ; // Tabu list
6   Identify  $A(s, k)$ ; // Aspiration set
7    $V \leftarrow N(s, k) \subseteq N(s) - T(s, k) + A(s, k)$ ;
8    $s \leftarrow$  best in  $V$ ;
9   if  $f(s) < f(s^*)$  then
10    |  $s^* \leftarrow s$ ;
11  end
12   $k++$ ;
13 end

```

Definition 7. The belief base $S_{m_1}^{\text{logic}}$, along with the belief bases detected in the preprocessing stage will conform the knowledge profile $\mathcal{E} = S_{m_1}^{\text{logic}} \cup \{K_1, \dots, K_n\}$.

Logical evaluator. This component evaluates the models of the belief profile. It builds a truth table of size $|\mathcal{P}(\mathcal{E})| \times 2^{|\mathcal{P}(\mathcal{E})|}$.

To ease the evaluation of formulas, we transform each belief base into its RPN (*Reverse Polish Notation*). We include the RPN using Dijkstra's *Shunting-yard algorithm* implemented for logical operators [3] using the operators in ρ with the usual precedence. Where operators have equal precedence, parenthesis along with their associativity indicates in which order they will be applied.

Belief merging operator. Algorithm 2 presents an implementation of the Δ_{ps} operator using the notion of Partial Satisfiability, which is defined as follows:

Definition 8. *Partial Satisfiability* is defined over a belief base normalized in DNF format: Q_K , any interpretation $w \in \mathcal{W}$ and $|\mathcal{P}| = n$. The *Partial Satisfiability* of Q_K for w , denoted by $w_{ps}(Q_K)$, is defined as:

1. If Q_K is a conjunction $C_1 \wedge \dots \wedge C_s$ where each C_i is a literal, then:

$$w_{ps}(Q_K) = \max \left\{ \sum_{i=1}^s \frac{w(C_i)}{s}, \frac{n - |\mathcal{P}(\bigwedge_{i=1}^s C_i)|}{2n} \right\} \quad (1)$$

where:

n is the number of atoms of the considered language.

s is the number of literals appearing in the conjunction.

Algorithm 2: Δ_{ps} (*PS-Merge*)

Input : V : Number of variables of E
 B : Number of bases of E
 D : Vector of number of disjuncts of each base in E
 L : Matrix of occurrences of literals in each disjunct of each base of E

Output: *Solution_Set*: The set of models of $\Delta_{ps}(E)$

```
1 Solution  $\leftarrow \emptyset$ 
2 Max-Sum  $\leftarrow 0$ 
3  $W \leftarrow$  Matrix whose rows are all the possible interpretations for  $V$  variables.
4 for  $s = 1 \dots B$  do
5   |  $ID_s \leftarrow \sum_{k=1}^{s-1} D_{k+1}$ 
6 end
7 for  $i = 1 \dots 2^V$  do
8   | Sum  $\leftarrow 0$ 
9   | for  $s = 1 \dots B$  do
10    | ps-disjunct  $\leftarrow \emptyset$ 
11    | for  $d = ID_s \dots ID_s + D_s$  do
12    |   | satisfied  $\leftarrow 0$ 
13    |   | conjuncts  $\leftarrow 0$ 
14    |   | vars-not-appearing  $\leftarrow 0$ 
15    |   | for  $j = 1 \dots V$  do
16    |   |   | if  $W_{i,j} = 1$  then
17    |   |   |   | satisfied  $\leftarrow$  satisfied +  $L_{d,2j-1}$ 
18    |   |   |   end
19    |   |   | if  $W_{i,j} = 0$  then
20    |   |   |   | satisfied  $\leftarrow$  satisfied +  $L_{d,2j}$ 
21    |   |   |   end
22    |   |   | conjuncts  $\leftarrow$  conjuncts +  $L_{d,2j-1} + L_{d,2j}$ 
23    |   |   | if  $L_{d,2j} = 0$  and  $L_{d,2j-1} = 0$  then
24    |   |   |   | vars-not-appearing  $\leftarrow$  vars-not-appearing+1
25    |   |   |   end
26    |   |   end
27    |   | ps-disjunct  $\leftarrow$  ps-disjunct  $\cup \{ \max(\frac{\textit{satisfied}}{\textit{conjuncts}}, \frac{\textit{vars-not-appearing}}{2^V}) \}$ 
28    |   | end
29    |   | PS  $\leftarrow$  max(ps - disjunct)
30    |   | Sum  $\leftarrow$  Sum + PS
31    |   end
32    | if Sum > MaxSum then
33    |   | Solution  $\leftarrow \{i\}$ 
34    |   | MaxSum  $\leftarrow$  Sum
35    |   end
36    | else if Sum = MaxSum then
37    |   | Solution  $\leftarrow$  Solution  $\cup \{i\}$ 
38    |   end
39 end
40 Solution_Set = {ith row of  $W$  |  $i \in$  Solution}
```

2. If Q_K is a disjunction $D_1 \vee \dots \vee D_r$ where each D_i is a conjunction of literals, then:

$$w_{ps}(Q_K) = \max \{w_{ps}(D_1), \dots, w_{ps}(D_r)\} \quad (2)$$

Definition 9. The following pre-order is defined:

$$w \leq_{ps}^E w' \text{ iff } \sum_{i=1}^m w_{ps}(Q_{K_i}) \leq \sum_{i=1}^m w'_{ps}(Q_{K_i}) \quad (3)$$

Definition 10. Δ_{ps} operator is defined by its models as:

$$mod(\Delta_{ps}(E)) = \max(\mathcal{W}, \leq_{ps}^E) \quad (4)$$

As mentioned, Δ_{ps} belongs to the majority operators subclass. However, by computing the minimum value from the Partial Satisfiability of the belief bases, it tends to be an arbitration operator. With this refinement, we can choose the resulting interpretations impartially, satisfying each stakeholder as much as possible.

4 Motivating example

Many real-world COPs may include both numerical and symbolic issues. To illustrate our proposal, we now introduce the Course Timetabling Problem (CTP). In general, CTP consists in the allocation of courses, groups of students and professors to time slots, subject to constraints of two types:

- Hard constraints: Are conditions that must be fully satisfied in order to obtain a feasible solution.
- Soft constraints: Are optional requirements that are desirable but not mandatory.

Hard constraints are considered requirements, while soft constraints are considered as preferences. As far as we know, there is no record of a proposal for solution or hybrid method using Belief Merging.

There are various formulations of CTP [18], the following is a variation of [19] considering the professors' teaching preferences.

Example 1. A school needs a timetable of q courses K_1, \dots, K_q , and each i course K_i consists of k_i lectures. There are t teachers P_1, \dots, P_t , which all can teach any lecture. There are r groups S_1, \dots, S_r of courses that have students in common. This means that the courses in S_l must be scheduled all at different times. The number of periods is p , and l_k is the maximum number of lectures that can be scheduled at period k (i.e., the number of rooms available at period k). It is recommended to consider each teacher expertise, i.e., previous courses taught, represented as a set H_j of courses $\{K_{i_1}, \dots, K_{i_{m_j}}\}$. There is also an optional list of courses preferences from each teacher, represented as a set F_j of courses $\{K_{i_1}, \dots, K_{i_{m_j}}\}$.

4.1 Module 1

The following formulation serves as an input for the first module of our framework, as it models the CTP as a search problem and therefore suitable for the Tabu Search implemented:

$$\text{find } S_{m_1} = \{x_{ijk} \mid i = 1, \dots, q; j = 1, \dots, t; k = 1, \dots, p\} \quad (5)$$

subject to the following constraints:

$$\sum_{k=1}^p x_{ijk} = k_i \quad (i = 1, \dots, q; j = 1, \dots, t) \quad (6)$$

$$\sum_{i=1}^q x_{ijk} \leq l_k \quad (j = 1, \dots, t; k = 1, \dots, p) \quad (7)$$

$$\sum_{i \in S_l} x_{ijk} \leq 1 \quad (l = 1, \dots, r; j = 1, \dots, t; k = 1, \dots, p) \quad (8)$$

$$x_{ijk} \in \{0, 1\} \text{ with } (i = 1, \dots, q; j = 1, \dots, t; k = 1, \dots, p) \quad (9)$$

S_{m_1} , in this case, is a timetable where $x_{ijk} = 1$ if a lecture of course K_i given by teacher P_j is scheduled at period k , and $x_{ijk} = 0$ otherwise. Constraint 6 impose that each course is composed of the correct number of lectures. Constraint 7 enforce that at each time there aren't more lectures than rooms. Constraint 8 prevent conflicting lectures to be scheduled during the same period. These are the hard constraints of the problem.

Soft constraints are about the courses preferences. A pure optimization strategy usually would address this soft constraint adding a weight variable d_{ij} representing the desirability of courses by each teacher in a maximization objective function. However, we will leave this task to the second module of our framework.

4.2 Module 2

A belief merging operator computes models of logic formulas, so this component creates the belief bases by translating the crucial sets to make the fusion. Each belief base represents a propositional formula where each element φ in the original set is represented by the logical variable $\hat{\varphi}$. This component translates the following sets:

1. The timetable S_{m_1} generated by Module 1 into a knowledge bases format:

$$S_{m_1, j}^{\text{logic}} = \bigwedge \hat{x}_{ijk} \quad (\forall x_{ijk} \in S_{m_1} \mid x_{ijk} = 1; i = 1, \dots, q; k = 1, \dots, p) \quad (10)$$

2. The set of preferential courses into a knowledge base format:

$$F_j^{\text{logic}} = \bigwedge \hat{K} \quad (\forall K \in F_j) \quad (11)$$

3. A similar translation for the historical courses:

$$H_j^{\text{logic}} = \bigwedge \hat{K} \quad (\forall K \in H_j) \quad (12)$$

In order to fusion the desires and expertise into the current timetable, we obtain the t independent belief profiles to merge, one for each teacher:

$$\mathcal{E}_j = \left(S_{m_1, j}^{\text{logic}}, F_j^{\text{logic}}, H_j^{\text{logic}} \right) \quad (13)$$

Δ_{ps} evaluates the models of the belief profiles. The output will be the best configuration of courses for each teacher, according to the soft constraints.

5 Conclusion

We present the preliminary ideas on the development of a hybrid framework for Constraint Optimization Problems, taking advantages of two different techniques such as Tabu Search and a Belief Merging operator.

We believe that this two-fold approach may produce more fair solutions in problems where a consensus is needed, because that is precisely what Belief Merging promotes, to satisfy most of the contradictory opinions.

Further work considers a full implementation of our proposal and tests over real-world scenarios.

References

1. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **35**(3), 268–308 (2003). <https://doi.org/10.1145/937503.937505>
2. Brownlee, J.: *Clever algorithms : nature-inspired programming recipes*. Lulu, Online (2011)
3. Chávez-Bosquez, O., Pozos-Parra, P., McAreavey, K.: On the development of a logic calculator: a novel tool to perform logical operations. In: Osorio Galindo, M., Marcial Romero, J., Zepeda Cortés, C., Olmos Pineda, I. (eds.) *Tenth Latin American Workshop on Logic/Languages, Algorithms and New Methods of Reasoning*. pp. 9–16. CEUR Workshop Proceedings, Puebla, México (2016), <http://ceur-ws.org/Vol-1659/paper2.pdf>
4. Dubois, D., Liu, W., Ma, J., Prade, H.: The basic principles of uncertain information fusion. an organised review of merging rules in different representation frameworks. *Information Fusion* **32**, 12–39 (2016). <https://doi.org/http://dx.doi.org/10.1016/j.inffus.2016.02.006>
5. Gendreau, M.: *Handbook of Metaheuristics*, chap. An Introduction to Tabu Search, pp. 37–54. Springer US, Boston, MA (2003). https://doi.org/10.1007/0-306-48056-5_2
6. Glover, F.: Tabu Search: A Tutorial. *Interfaces* **20**(4), 21 (1990). <https://doi.org/10.1287/inte.20.4.74>
7. Glover, F., Laguna, M.: *Tabu Search*. Kluwer Academic Publishers, USA, 1st edn. (1997)

8. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* **13**(5), 533 – 549 (1986). [https://doi.org/https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/https://doi.org/10.1016/0305-0548(86)90048-1), applications of Integer Programming
9. Kareem, S., Pozos-Parra, P., Wilson, N.: An application of belief merging for the diagnosis of oral cancer. *Applied Soft Computing* (2017). <https://doi.org/http://dx.doi.org/10.1016/j.asoc.2017.01.055>
10. Konieczny, S., Pino Pérez, R.: Logic based merging. *Journal of Philosophical Logic* **40**(2), 239–270 (2011)
11. Konieczny, S., Prez, R.P.: Merging information under constraints: A logical framework. *Journal of Logic and Computation* **12**(5), 773–808 (2002). <https://doi.org/10.1093/logcom/12.5.773>
12. Luke, S.: *Essentials of Metaheuristics*. Lulu, second edn. (2013), available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>
13. McAreavey, K., Liu, W., Miller, P.: Computational approaches to finding and measuring inconsistency in arbitrary knowledge bases. *International Journal of Approximate Reasoning* **55**(8), 1659–1693 (2014). <https://doi.org/10.1016/j.ijar.2014.06.003>
14. Michalewicz, Z., Fogel, D.: *How to Solve It: Modern Heuristics*. Springer-Verlag Berlin Heidelberg, 2 edn. (2004). <https://doi.org/10.1007/978-3-662-07807-5>
15. Pigozzi, G.: Belief merging and judgment aggregation. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edn. (2016)
16. Pozos Parra, P., Borja Macías, V.: Partial satisfiability-based merging. In: Gelbukh, A., Kuri Morales, Á. (eds.) *MICAI 2007: Advances in Artificial Intelligence*. 6th Mexican International Conference on Artificial Intelligence. *Lecture Notes in Computer Science*, vol. 4827, pp. 225–235. Springer Berlin Heidelberg (2007). https://doi.org/10.1007/978-3-540-76631-5_22
17. Pozos-Parra, P., Chávez-Bosquez, O., McAreavey, K.: Merginator: A belief merging tool for consensus support. *Journal of Intelligent & Fuzzy Systems* **34**(5), 3199–3210 (2018)
18. Schaerf, A.: A survey of automated timetabling. *Artificial Intelligence Review* **13**(2), 87–127 (1999). <https://doi.org/10.1023/A:1006576209967>
19. de Werra, D.: An introduction to timetabling. *European Journal of Operational Research* **19**(2), 151–162 (1985). [https://doi.org/10.1016/0377-2217\(85\)90167-5](https://doi.org/10.1016/0377-2217(85)90167-5)