Generalized Labeling Problems with a Majority Polymorphism for a Certain Class of Semirings

Vodolazskiy Evgeniy

Department of Image Processing and Recognition
IRTC ITS
waterlaz@gmail.com

Abstract

The article describes a generalization of a constraint satisfaction problem and its max-min relaxation. The general problem is defined in terms of a commutative semiring and considers a special case of semirings with idempotent operations. The concept of polymorphism is generalized for this special case and the polynomial time algorithm for problems with a majority polymorphism is described.

1. Introduction

This article generalizes the idea of using polymorphisms to define tractable subclasses of NP-complete problems [1] and is a direct generalization of algorithms and theorems for max-min labeling problems [2] to commutative semirings with idempotent operations.

2. Notation

The article talks about commutative semirings $(\oplus, \otimes, S)$. The symbol $\otimes$ is omitted in most of the formulas, so instead of $a \otimes b$ we use ab. When writing addition or multiplication operations over some sets, the set may be omitted. For instance, one might find the expression $\bigotimes\limits_{x}$ instead of $\bigotimes\limits_{x \in X}$ in the article. Hopefully, the set is obvious from the context.

3. Problem formulation

A lot of problems can be formulated in terms of a general labeling problem on a commutative semiring.

Definition 1. *The general labeling problem on a commutative semiring* $(\oplus, \otimes, S)$ *is a quadruple*

$$\Big\langle X, T, \mathcal{T} \subset 2^T, \big(f_{T'} : X^{T'} \to S | T' \in \mathcal{T}\big) \Big\rangle,$$

where T is a finite set of variables, X is a finite set of variables' values, $f_{T'}$, $T' \in \mathcal{T}$, are functions. It is necessary to compute the value $\bigoplus\limits_{\overline{x} \in X^k} \bigotimes\limits_{T' \in \mathcal{T}} f_{T'}(\overline{x}(T'))$ *for a given problem.*

There are two practical special cases of this problem: a constraint satisfaction problem and a max-min labeling problem.

3.1. Constraint satisfaction problem

A constraint satisfaction problem[3, 4] is a labeling problem on a semiring $\big(\bigvee, \&, \{0, 1\}\big)$. Functions $f_{T'}$ define constraints on subsets of variables T'. Computing $\bigoplus\limits_{\overline{x} \in X^k} \bigotimes\limits_{T' \in \mathcal{T}} f_{T'}(\overline{x}(T')) = \bigvee\limits_{\overline{x} \in X^k} \mathop{\&}\limits_{T' \in \mathcal{T}} f_{T'}(\overline{x}(T'))$ is equivalent to determining if one can assign values $\overline{x}^*$ to variables that fulfill all the constraints $f_{T'}(\overline{x}^*(T')) = 1, T' \in \mathcal{T}$. In other words, it is necessary to solve a system of Boolean equations. The values $\overline{x}^*$ are called a solution to the constraint satisfaction problem. Note that a general labeling problem in Definition 1 does not define a concept of a solution.

3.2. A max-min labeling problem

Sometimes a constraint satisfaction problem may not have a solution. In this case one might want to find some sort of relaxed solution by relaxing the constraints so that each constraint has a grade of satisfiability. The solution to a max-min labeling problem are such variables' values $\overline{x}^*$ that make all the constraints as much satisfied as possible [2]. In other words, to find such $\overline{x}^*$ that maximizes the value $\min\limits_{T' \in \mathcal{T}} f_{T'}(\overline{x}(T'))$. This is equivalent to solving the labeling problem on a semiring $(\max, \min, \mathbb{R})$.

Both the constraint satisfaction problem and the max-min labeling problem are NP-complete. We will show a tractable subclass of the general labeling problem for commutative semirings $(\oplus, \otimes, S)$ with idempotent operations $\oplus$ and $\otimes$, that is $a \oplus a = a$ and $aa = a$. Evidently, both $\big(\bigvee, \&, \{0, 1\}\big)$ and $(\max, \min, \mathbb{R})$ fulfill these rules.

The tractable subclass of problems will be defined in the following sections using the concept of polymorphism.

4. Polymorphisms and invariants

Throughout the paper we talk about functions $f_{T'} : X^{T'} \to S$ that are sometimes called constraints. In order to define the concept of polymorphisms we introduce the functions $p : X \times X \times X \to X$ that we call operators. Purely for convenience purposes we extend the operators on triples of values $p : X \times X \times X \to X$ to operators on triples of tuples $p : X^k \times X^k \times X^k \to X^k$ by applying the operator p element-wise.

$$p(\overline{x}, \overline{y}, \overline{z}) = \\ = \big(p(x_1, y_1, z_1), p(x_2, y_2, z_2), \ldots, p(x_k, y_k, z_k)\big),$$

Definition 2. *An operator $p : X \times X \times X \to X$ is a polymorphism of a function $f : X^k \to S$ (or f is invariant by p) if and only if the equality*

$$f(\overline{x})f(\overline{y})f(\overline{z}) \oplus f\big(p(\overline{x}, \overline{y}, \overline{z})\big) = f\big(p(\overline{x}, \overline{y}, \overline{z})\big) \qquad (1)$$

holds for all $\overline{x}, \overline{y}, \overline{z} \in X^k$.

The set of all polymorphisms of a function f is denoted by $\mathrm{Pol}\{f\}$. The set of all functions invariant by p is denoted by $\mathrm{Inv}\{p\}$.

If there is an operator p such that p is a polymorphism of all problem's constraint functions then we say that this problem has a polymorphism p.

The Definition 2 is a direct generalization of the classical definition of a polymorphism [1, 3, 5, 6] defined for a semiring $\big(\bigvee, \&, \{0, 1\}\big)$. Indeed, the expression (1) becomes $f(\overline{x}) \,\&\, f(\overline{y}) \,\&\, f(\overline{z}) \vee f\big(p(\overline{x}, \overline{y}, \overline{z})\big) = f\big(p(\overline{x}, \overline{y}, \overline{z})\big)$ and is equivalent to $f(\overline{x}) \,\&\, f(\overline{y}) \,\&\, f(\overline{z}) \to f\big(p(\overline{x}, \overline{y}, \overline{z})\big)$.

Polymorphisms on semirings with idempotent operations have two remarkable properties as shown by Lemma 1 and Lemma 2.

Lemma 1. *Let $f : X^k \to S$ and $g : X^k \to S$ be two functions invariant by the same operator p. Then the function $(f \otimes g)(\overline{x}) = f(\overline{x}) \otimes g(\overline{x})$ is also invariant by the operator p.*

Proof.

$$\begin{aligned}
& f(p(\overline{x}, \overline{y}, \overline{z}))g(p(\overline{x}, \overline{y}, \overline{z})) = \\
&= \Big[f(\overline{x})f(\overline{y})f(\overline{z}) \oplus f(p(\overline{x}, \overline{y}, \overline{z}))\Big] \otimes && (2)\\
&\quad \otimes \Big[g(\overline{x})g(\overline{y})g(\overline{z}) \oplus g(p(\overline{x}, \overline{y}, \overline{z}))\Big] = \\
&= \Big[f(\overline{x})f(\overline{y})f(\overline{z}) \oplus f(p(\overline{x}, \overline{y}, \overline{z}))\Big] \otimes && (3)\\
&\quad \otimes \Big[g(\overline{x})g(\overline{y})g(\overline{z}) \oplus g(p(\overline{x}, \overline{y}, \overline{z}))\Big] \oplus \\
&\quad \oplus f(\overline{x})g(\overline{x})f(\overline{y})g(\overline{y})f(\overline{z})g(\overline{z}) = \\
&= \big[f(\overline{x})g(\overline{x})\big]\big[f(\overline{y})g(\overline{y})\big]\big[f(\overline{z})g(\overline{z})\big] \oplus && (4)\\
&\quad \oplus f(p(\overline{x}, \overline{y}, \overline{z}))g(p(\overline{x}, \overline{y}, \overline{z})).
\end{aligned}$$

Equation (2) is obtained by applying (1). If one opens the brackets in (2) one of the summands will be $f(\overline{x})g(\overline{x})f(\overline{y})g(\overline{y})f(\overline{z})g(\overline{z})$. Therefore, we can add this summand once more and it won't change the value of the sum due to idempotency of $\oplus$. Finally, we obtain (4) by applying (1) once again. □

Definition 3. *A function $f_{X^n} : X^n \to S$ is called a projection of a function $f : X^n \times X^m \to S$ into n variables if and only if $f_{X^n}(\overline{x}) = \bigoplus_{\overline{y} \in X^m} f(\overline{x}, \overline{y})$.*

Lemma 2. *If a function $f : X^n \times X^m \to S$ is invariant by some operator p then its projection to n variables $g : X^n \to S$ is also invariant by p.*

Proof.

$$\begin{aligned}
& \Big[\bigoplus_{\overline{t}_1} f(\overline{x}, \overline{t}_1)\Big]\Big[\bigoplus_{\overline{t}_2} f(\overline{y}, \overline{t}_2)\Big]\Big[\bigoplus_{\overline{t}_3} f(\overline{z}, \overline{t}_3)\Big] \oplus \\
&\quad \oplus \bigoplus_{\overline{t}} f(p(\overline{x}, \overline{y}, \overline{z}), \overline{t}) = \\
&= \bigoplus_{\overline{t}_1}\bigoplus_{\overline{t}_2}\bigoplus_{\overline{t}_3} f(\overline{x}, \overline{t}_1)f(\overline{y}, \overline{t}_2)f(\overline{z}, \overline{t}_3) \oplus && (5)\\
&\quad \oplus \bigoplus_{\overline{t}} f(p(\overline{x}, \overline{y}, \overline{z}), \overline{t}) = \\
&= \bigoplus_{\overline{t}_1}\bigoplus_{\overline{t}_2}\bigoplus_{\overline{t}_3} \Big[f(\overline{x}, \overline{t}_1)f(\overline{y}, \overline{t}_2)f(\overline{z}, \overline{t}_3) \oplus && (6)\\
&\qquad \oplus f\big(p(\overline{x}, \overline{y}, \overline{z}), p(\overline{t}_1, \overline{t}_2, \overline{t}_3)\big)\Big] \oplus \\
&\quad \oplus \bigoplus_{\overline{t}} f(p(\overline{x}, \overline{y}, \overline{z}), \overline{t}) = \\
&= \bigoplus_{\overline{t}_1}\bigoplus_{\overline{t}_2}\bigoplus_{\overline{t}_3} f\big(p(\overline{x}, \overline{y}, \overline{z}), p(\overline{t}_1, \overline{t}_2, \overline{t}_3)\big) \oplus && (7)\\
&\quad \oplus \bigoplus_{\overline{t}} f(p(\overline{x}, \overline{y}, \overline{z}), \overline{t}) = \\
&= \bigoplus_{\overline{t}} f(p(\overline{x}, \overline{y}, \overline{z}), \overline{t}). && (8)
\end{aligned}$$

In equation (5) we simply opened brackets. Since the summand $f\big(p(\overline{x}, \overline{y}, \overline{z}), p(\overline{t}_1, \overline{t}_2, \overline{t}_3)\big)$ is already present in the sum $\bigoplus_{\overline{t}} f(p(\overline{x}, \overline{y}, \overline{z}), \overline{t})$ for any $\overline{t}_1, \overline{t}_2, \overline{t}_3 \in X$ we can add it once more and the equality (6) will be valid due to $\oplus$ being idempotent. Since p is a polymorphism of f, equality (7) holds. Equality (8) is valid for the same reasons (6) is. □

5. Majority polymorphism

We consider a special type of polymorphisms called majority polymorphism.

Definition 4. *An operator $p : X \times X \times X \to X$ is called a majority operator if and only if for any $x, y \in X$*

$$p(x, x, y) = p(x, y, x) = p(y, x, x) = x. \qquad (9)$$

Majority polymorphisms on a semiring with idempotent operations have one very useful property.

Lemma 3. *Any function f of three or more variables invariant by a majority operator p can be expressed as a product of three functions of less variables. Moreover, these three functions are also invariant by p.*

Proof. We prove that any function $f(\overline{x},\overline{y},\overline{z})$ of three sets of variables $\overline{x}\in X^k, \overline{y}\in X^m, \overline{z}\in X^n$ can be expressed as a product of three projections.

$$f(\overline{x},\overline{y},\overline{z}) = = \bigoplus_{\overline{x}_1\in X^k} f(\overline{x}_1,\overline{y},\overline{z}) \bigoplus_{\overline{y}_1\in X^m} f(\overline{x},\overline{y}_1,\overline{z}) \bigoplus_{\overline{z}_1\in X^n} f(\overline{x},\overline{y},\overline{z}_1). \quad (10)$$

Indeed,

$$\bigoplus_{\overline{x}_1} f(\overline{x}_1,\overline{y},\overline{z}) \bigoplus_{\overline{y}_1} f(\overline{x},\overline{y}_1,\overline{z}) \bigoplus_{\overline{z}_1} f(\overline{x},\overline{y},\overline{z}_1) =$$

$$= \bigoplus_{\overline{x}_1}\bigoplus_{\overline{y}_1}\bigoplus_{\overline{z}_1} f(\overline{x}_1,\overline{y},\overline{z}) f(\overline{x},\overline{y}_1,\overline{z}) f(\overline{x},\overline{y},\overline{z}_1) =$$

$$= \bigoplus_{\overline{x}_1}\bigoplus_{\overline{y}_1}\bigoplus_{\overline{z}_1} \Big[f(\overline{x}_1,\overline{y},\overline{z}) f(\overline{x},\overline{y}_1,\overline{z}) f(\overline{x},\overline{y},\overline{z}_1) \oplus \quad (11)$$

$$\oplus f\big(p(\overline{x}_1,\overline{x},\overline{x}), p(\overline{y},\overline{y}_1,\overline{y}), p(\overline{z},\overline{z},\overline{z}_1)\big)\Big] =$$

$$= \bigoplus_{\overline{x}_1}\bigoplus_{\overline{y}_1}\bigoplus_{\overline{z}_1} f\big(p(\overline{x}_1,\overline{x},\overline{x}), p(\overline{y},\overline{y}_1,\overline{y}), p(\overline{z},\overline{z},\overline{z}_1)\big) = \quad (12)$$

$$= \bigoplus_{\overline{x}_1}\bigoplus_{\overline{y}_1}\bigoplus_{\overline{z}_1} f(\overline{x},\overline{y},\overline{z}) = f(\overline{x},\overline{y},\overline{z}) \quad (13)$$

Since p is a majority operator, the triple $\big(p(\overline{x}_1,\overline{x},\overline{x}), p(\overline{y},\overline{y}_1,\overline{y}), p(\overline{z},\overline{z},\overline{z}_1)\big)$ equals $(\overline{x},\overline{y},\overline{z})$. The summand $f(\overline{x},\overline{y},\overline{z})$ is already present in the sum. Therefore, we can add it one more time in the form of $f\big(p(\overline{x}_1,\overline{x},\overline{x}), p(\overline{y},\overline{y}_1,\overline{y}), p(\overline{z},\overline{z},\overline{z}_1)\big)$ and the sum will not change. Hence, equality (11) is valid. We obtain (12) by using (1). Finally, (13) is valid due to idempotency of addition. Note that due to Lemma 2 each of the three functions are invariant by p. □

Lemma 3 results in the following theorem.

Theorem 1. *Any function $f : X^k \to S$ invariant by a majority operator p can be expressed as a product of functions of two variables.*

$$f(\overline{x}) = \bigotimes_{i,j\in\overline{1,k}} f_{ij}(x_i,x_j), \quad f_{ij}(x_i,x_j) = \bigoplus_{\substack{\overline{y}:y_i=x_i\\ y_j=x_j}} f(\overline{y}).$$

Each of these functions are projections of f and are invariant by operator p.

Proof. According to Lemma 2 the function f can be broken into three functions of less variables. Since f is invariant by p each of these functions is also invariant by p and in turn can be broken into functions of less variables. We can repeat this process until we obtain functions of two variables invariant by the majority operator p. If there are more than one function that depends on the same pair of variables then these functions are replaced by their product according to Lemma 1. □

Theorem 1 allows to transform any given problem with a majority polymorphism into a problem with the same polymorphism with all constraint functions depending only on two variables.

6. Star to simplex transformation and variable exclusion

Similarly to [2], the main procedure of the algorithm to solve a general labeling problem with a majority polymorphism is a star to simplex transformation that works with constraint functions of two variables. Figure 1 shows an example of this transformation with variables shown as circles and constraint functions of two variables are shown as lines connecting these circles. A star is a set of variables with one center variable and a set of constraint functions such that each function depends on a center variable and some other variable. All variables that are not a center variable are called rays. A star to simplex transformation replaces these constraint functions by an equivalent set of functions that do not depend on a center variable.

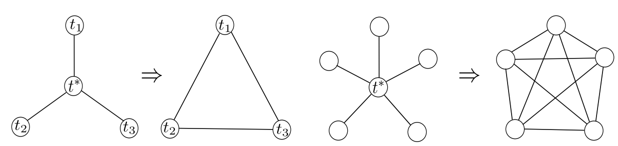

Figure 1: *Star to simplex transformation.*

Let a set of variables T and a center variable t^* form a star. The star to simplex transformation is a transformation of the set of constraints $f_{t^*t}, t\in T$, to an equivalent set of constraints $f_{t_1t_2},\ t_1,t_2\in T$. In other words

$$\bigoplus_{x^*}\bigotimes_{t\in T} f_{t^*t}(x^*,x(t)) = \bigotimes_{\substack{t_1\in T\\ t_2\in T}} f_{t_1t_2}(x(t_1),x(t_2)). \quad (14)$$

This is a direct generalization of a star to simplex transformation defined for a $(\max,\min,\mathbb{R})$ semiring [2].

Since $\bigoplus_{x^*}\bigotimes_{t\in T} f_{t^*t}(x^*,x(t))$ is a projection of $\bigotimes_{t\in T} f_{t^*t}(x^*,x(t))$, it is invariant by a majority operator according to Lemmas 1 and 2. Therefore, we can transform these constraint functions into $f_{t_1t_2}$ by choosing

$f_{t_1 t_2}$ to be projections of $\bigoplus\limits_{x^*} \bigotimes\limits_{t \in T} f_{t^* t}(x^*, x(t))$ to pairs of variables t_1, t_2 according to Theorem 1.

$$f_{t_1 t_2}(x(t_1), x(t_2)) = \\ = \bigoplus_{\overline{x} \in X^{T \setminus \{t_1, t_2\}}} \bigoplus_{x^*} \bigotimes_{t \in T} f_{t^* t}(x^*, x(t)). \quad (15)$$

Let us denote $\varphi_t(x^*) = \bigoplus\limits_{x \in X} f_{t^* t}(x^*, x)$. Then (15) becomes

$$f_{t_1 t_2}(x(t_1), x(t_2)) = \\ = \bigoplus_{x^*} \bigotimes_{\substack{t \neq t_1 \\ t \neq t_2}} \varphi_t(x^*) f_{t^* t_1}(x^*, x(t_1)) f_{t^* t_2}(x^*, x(t_2)) = \\ = \bigoplus_{x^*} \Big[f_{t^* t_1}(x^*, x(t_1)) f_{t^* t_2}(x^*, x(t_2)) \otimes \bigotimes_{\substack{t \neq t_1 \\ t \neq t_2}} \varphi_t(x^*) \Big].$$

All values $\varphi_t(x^*)$ can be computed in $O(|T||X|^2)$ time. A naive approach to computing each $f_{t_1 t_2}(x(t_1), x(t_2))$ requires $O(|T||X|)$ time or $O(|T|^3|X|^3)$ in total. Fortunately, a rather standard trick allows to reduce this time to $O(|T|^2|X|^3)$. Assume that the set T is ordered $T = \{t_1, t_2, \ldots, t_n\}$ (the specific order is irrelevant). $\bigotimes\limits_{\substack{t \neq t_a \\ t \neq t_b}} \varphi_t(x^*) =$

$$= \bigotimes_{i=1}^{a-1} \varphi_{t_i}(x^*) \otimes \bigotimes_{i=a+1}^{b-1} \varphi_{t_i}(x^*) \otimes \bigotimes_{i=b+1}^{n} \varphi_{t_i}(x^*) = \\ = q(1, a-1, x^*) \otimes q(a+1, b-1, x^*) \otimes q(b+1, n, x^*),$$

where $q(a, b, x^*) = \bigotimes\limits_{i=a}^{b} \varphi_{t_i}(x^*)$ and all q's can be computed in $O(|T|^2|X|)$ by using dynamic programming. Then it takes only $O(|X|)$ time to compute each f.

The star to simplex transformation produces a constraint function invariant by a majority operator for each pair of variables. The procedure of excluding a variable from the problem proceeds as follows:

1. Choose a variable and all the constraint functions that depend on this variable. This variable will become the center of the star. All other variables that the chosen functions depend from are the rays of the star.
2. Use the star to simplex transformation to replace the chosen restriction functions with functions that depend only on the rays.
3. If the original problem had a restriction function for any two rays then multiply it by the function obtained with a star to simplex transformation. Otherwise, just use the function from the transformation.

The algorithm for solving a general labeling problem proceeds by excluding one variable at a time until the problem has three or less variables. Let these variables be numbered $1, 2, 3$ and let f_{12}, f_{13}, f_{23} be constraint functions over these variables. Then the answer is obtained by directly computing the formula $\bigoplus\limits_{x_1 \in X} \bigoplus\limits_{x_2 \in X} \bigoplus\limits_{x_3 \in X} f_{12}(x_1, x_2) f_{13}(x_1, x_3) f_{23}(x_2, x_3)$. Since the exclusion procedure is performed $O(|T|)$ times, the algorithm computes $\bigoplus\limits_{\overline{x} \in X^T} \bigotimes\limits_{T' \in \mathcal{T}} f_{T'}(\overline{x}(T'))$ for any given problem with a majority polymorphism in $O(|T|^3|X|^3)$ time.

Note that determining whether some problem has a majority polymorphism may turn out to be a complex task. The advantage of the proposed algorithm is that one does not have to know the polymorphism. If the polymorphism exists, the algorithm computes the correct value $\bigoplus\limits_{\overline{x} \in X^T} \bigotimes\limits_{T' \in \mathcal{T}} f_{T'}(\overline{x}(T'))$ without knowing or computing the polymorphism itself.

7. Conclusions

A generalized labeling problem on a commutative semiring with idempotent operations (both multiplication and addition) that has a majority polymorphism can be reduced to a problem with the same polymorphism and restriction functions that depend only on two variable. The latter problem can be solved in $O(|T|^3|X|^3)$ time by the proposed algorithm.

8. References

[1] A. Bulatov and P. Jeavons, "Tractable constraints closed under a binary operation" Technical Report PGR-TR-12-00 (Oxford University Computing Laboratory, Oxford, 2000).

[2] Vodolazskii, E.V., Flach, B. and Schlesinger, M.I., "Minimax problems of discrete optimization invariant under majority operators", Computational Mathematics and Mathematical Physics, Num. 8, Vol. 54:1327-1336, 2014.

[3] F. Rossi, P. van Beek, and T. Walsh, "Handbook of Constraint Programming" (Elsevier Science, New York, 2006).

[4] O. A. Shcherbina, "Constraint satisfaction and constraint programming" Intellekt. Sist. 15(14), 53170 (2011).

[5] A. Bulatov, "Tractable conservative constraint satisfaction problems" Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science LICS03 (Washington, DC, USA, 2003).

[6] A. Bulatov, "Complexity of conservative constraint satisfaction problems" ACM Trans. Comput. Logic 12(4), 24:124:66, July (2011)