Stop condition for subgradient minimization in dual relaxed (max,+) problem

Michail Schlesinger, Evgeniy Vodolazskiy, Nikolai Lopatka

International Research & Training Center of Information Technologies and Systems of National Academy of Sciences of Ukraine
schles@irtc.org.ua

Abstract. Subgradient descent methods for minimization of dual linear relaxed labeling problem are analysed. They are guaranteed to converge to the quality of the optimal relaxed labeling, but do not obtain an optimal relaxed labeling itself. Moreover, no stop condition is known for these methods upto now. The stop condition is defined and experimentally compared with the commonly-used stop conditions. The stop condition is defined in a way that when fulfilled a relaxed labeling is simultaneously obtained with arbitrary non-zero difference from the optimal labeling.

Key words: Labeling problem, relaxation, equivalent transformation, reparametrisation, subgradient descent, energy minimization

1 Introduction: definition of main concepts

Let T be a finite set of pixels and K be a finite set of labels. Let a function $\overline{k}: T \to K$ be called a labeling that for each pixel $t \in T$ defines a label $k(t) \in K$. The labeling $\overline{k}$ will be also referred to as a strict labeling as opposed to relaxed labeling defined below.

Let $\mathcal{T}$ be a set of unordered pairs $(t, t') \in \mathcal{T}$ of pixels that are called neighbours. The set $\mathcal{T}$ is symmetric and irreflexive in a sense that $(t, t') \in \mathcal{T} \Leftrightarrow (t', t) \in \mathcal{T}$ for any two pixels and $(t, t) \notin \mathcal{T}$. We will often use the notation $tt' \in \mathcal{T}$ instead of $(t, t') \in \mathcal{T}$ for simplicity. The subset $N(t) \subset T = \{t' | tt' \in \mathcal{T}\}$ is a set of neighbours for the pixel t. So the following three expressions are equivalent:

$$t' \in N(t) \Leftrightarrow t \in N(t') \Leftrightarrow tt' \in \mathcal{T} \ . \tag{1}$$

An ordered pair $(t, k), t \in T, k \in K$, will be called a vertex. For each neighbouring pair $tt' \in \mathcal{T}$ of pixels an unordered pair of vertices $((t, k), (t', k'))$ will be called an edge. We will say that a vertex (t^*, k^*) belongs to a labeling $\overline{k}$ if $k(t^*) = k^*$. An edge $((t^*, k^*), (t^{**}, k^{**}))$ belongs to a labeling $\overline{k}$ if both vertices (t^*, k^*) and (t^{**}, k^{**}) belong to $\overline{k}$.

Let $q(t, k)$ be a real number defined for each vertex (t, k) and called a vertex quality. Let $g((t, k), (t', k'))$ be a real number defined for each edge $((t, k), (t', k'))$

and called an edge quality. The quality $G(\overline{k})$ of a labeling $\overline{k}$ is defined as the sum of qualities of all edges and vertices that belong to the labeling:

$$G(\overline{k}) = \sum_{tt' \in \mathcal{T}} g\big((t, k(t)), (t', k(t'))\big) + \sum_{t \in T} q\big(t, k(t)\big) \ . \tag{2}$$

The strict labeling problem consists in finding the labeling with the best quality:

$$\overline{k^*} = \arg\max_{\overline{k} \in K^T} G(\overline{k}) \ . \tag{3}$$

The problem is known to be NP-complete. The so-called relaxed modification of this problem is much more simpler. It is based on the following concepts.

Let $\alpha(t,k), t \in T, k \in K$ be a real number called the weight of a vertex (t,k) and $\beta((t,k),(t',k')), tt' \in \mathcal{T}, k \in K, k' \in K$ be a real number called the weight of an edge $((t,k),(t',k'))$. Let α be an array $\big(\alpha(t,k) | t \in T, k \in K\big)$ of vertex weights and β be an array $\big(\beta((t,k),(t',k')) | tt' \in \mathcal{T}, k \in K, k' \in K\big)$ of edge weights. The pair (α, β) is called a weight function. A weight function (α, β) is called a relaxed labeling if it satisfies the following conditions:

$$\begin{cases} \alpha(t,k) = \sum\limits_{k' \in K} \beta\big((t,k),(t',k')\big), & t \in T, k \in K, t' \in N(t) \ ; \\ \sum\limits_{k \in K} \alpha(t,k) = 1, & t \in T \ ; \\ \beta\big((t,k),(t',k')\big) \geq 0, & tt' \in \mathcal{T}, k \in K, k' \in K \ . \end{cases} \tag{4}$$

The quality of a relaxed labeling is defined as follows:

$$G(\alpha,\beta) = \sum_{t \in T} \sum_{k \in K} \alpha(t,k) \cdot q(t,k) + \sum_{tt' \in \mathcal{T}} \sum_{k \in K} \sum_{k' \in K} \beta((t,k),(t',k')) \cdot g((t,k),(t',k')) \ . \tag{5}$$

The relaxed labeling problem consists in finding the relaxed labeling (α^*, β^*) with the best quality:

$$(\alpha^*, \beta^*) = \arg\max_{(\alpha,\beta)} G(\alpha, \beta) \ . \tag{6}$$

One can see that by restricting the $\alpha(t,k)$ and $\beta((t,k),(t',k'))$ to integer values the problem (6) becomes equivalent to the strict problem (3). In this case $\alpha(t,k') = 1$ in the relaxed problem means that $k(t) = k'$ in the strict problem.

2 Equivalent transformations of a relaxed labeling problem into trivial one

There exists an evident sufficient condition of the relaxed labeling optimality. If the relaxed labeling (α, β) fulfils the conditions

$$\begin{cases} q(t,k) < \max\limits_{l\in K} q(t,l) & \Rightarrow \alpha(t,k) = 0 \ , \\ g((t,k),(t',k')) < \max\limits_{l\in K, l'\in K} g((t,l),(t',l')) & \Rightarrow \beta((t,k),(t',k')) = 0 \ , \end{cases} \tag{7}$$

then it is the optimal relaxed labeling. Certainly, this condition is too strong and that is why the problems where it is satisfied are called trivial. The condition (7) can be weakened by using the concept of problem equivalent transformations also know as the re-parametrization ([6], [3]). Two relaxed labeling problems (q^1, g^1) and (q^2, g^2) defined on the same set of pixels and labels are called equivalent, $(q^1, g^1) \sim (q^2, g^2)$, if the quality of each relaxed labeling in one problem is equal to the quality of the same relaxed labeling in another. It is known that every relaxed labeling problem (q, g) has a trivial equivalent ([4]). This is the problem (q^*, g^*) that minimizes the value

$$P(q', g') = \sum_{tt'\in\mathcal{T}} \max_{k\in K, k'\in K} g'((t,k),(t',k')) + \sum_{t\in T} \max_{k\in K} q'(t,k) \ , \tag{8}$$

called power, in its equivalence class:

$$(q^*, g^*) = \arg \min_{(q',g')\sim(q,g)} P(q', g') \ . \tag{9}$$

Two labeling problems with qualities q_1, g_1 and q_2, g_2 respectively are equivalent if and only if there exist such numbers $\varphi_{tt'}(k), t \in T, t' \in N(t), k \in K$ that satisfy the equalities

$$\begin{cases} q_1(t,k) = q_2(t,k) - \sum\limits_{t'\in N(t)} \varphi_{tt'}(k), & t \in T, k \in K \ , \\ g_1((t,k),(t',k')) = g_2((t,k),(t',k')) + \varphi_{tt'}(k) + \varphi_{t't}(k'), & \\ & tt' \in \mathcal{T}, k \in K, k' \in K \ . \end{cases} \tag{10}$$

Therefore the power of equivalently transformed problem can be explicitly expressed as the function of φ,

$$\begin{aligned} P(\varphi) = & \sum_{tt'\in\mathcal{T}} \max_{k\in K, k'\in K} \left[g((t,k),(t',k')) + \varphi_{tt'}(k) + \varphi_{t't}(k')\right] + \\ & + \sum_{t\in T} \max_{k\in K} \left[q(t,k) - \sum_{t'\in N(t)} \varphi_{tt'}(k)\right] \ , \end{aligned} \tag{11}$$

and trivialization of the problem can be reduced to minimizing (11) with no restrictions on the variables φ. Well-known elegant message-passing method ([1]) as well as diffusion ([2]) attempt to decrease the power P but do not ensure its minimization ([3]).

3 Subgradient descent

The function P in (11) is a convex function of φ and therefore it's minimum can be found using the subgradient method ([3], [5]). Relative to the power minimization the subgradient method results in the following algorithm:
Initializaton

1. *define the sequence of numbers* γ_i *such that* $\lim_{i\to\infty} \gamma_i = 0$ *and* $\sum_{i=1}^{\infty} \gamma_i = \infty$ *;*
2. *for each edge assign input value* $g((t,k),(t',k'))$ *to qualities* $g^0((y,k),(t',k'))$ *;*
3. *for each vertex assign input value* $q(t,k)$ *to qualities* $q^0(t,k)$ *;*
4. *assign* $i = 0$ *;*

Iterations

1. *assign zero values to all* $\varphi_{tt'}(k), t \in T, t' \in N(t), k \in K$ *;*
2. *for each pair of neighbouring pixels* $tt' \in \mathcal{T}$ *choose a pair of labels* k *and* k' *such that* $g^i((t,k),(t',k')) = \max_{l\in K, l'\in K} g^i((t,l),(t',l'))$ *and assign*
   $\varphi_{tt'}(k) \ := \ \varphi_{tt'}(k) \ - \ 1;$
   $\varphi_{t't}(k') \ := \ \varphi_{t't}(k') \ - \ 1$ *;*
3. *for each pixel* $t \in T$ *choose a label* k *such that* $q^i(t,k) = \max_{l\in K} q^i(t,l)$ *and assign* $\varphi_{tt'}(k) := \varphi_{tt'}(k) + 1$ *for each object* $t' \in N(t)$ *;*
4. *for each edge* $((t,k),(t',k')), tt' \in \mathcal{T}, k \in K, k' \in K$ *calculate its new quality* $g^{i+1}((t,k),(t',k')) \ := \ g^i((t,k),(t',k')) \ + \ \varphi_{tt'}(k) \ + \ \varphi_{t't}(k')$ *;*
5. *for each vertex* $(t,k), t \in T, k \in K$ *calculate its new quality* $q^{i+1}(t,k) \ := \ q^i(t,k) \ - \sum_{t'\in N(t)} \varphi_{tt'}(k)$ *;*
6. *increment* $i := i + 1$ *and go to p.1 ;*

Let P^i be the power of the problem at the i-th iteration. Than according to the subgradient theory $\lim_{i\to\infty} P^i = \min_{\varphi} P(\varphi)$. So the above algorithm solves the power minimization problem, and therefore finds the quality of optimal relaxed labeling.

However, the algorithm requires a more precise definition. First of all, the stop condition is not specified. Second, it finds the quality of the optimal relaxed labeling but not the relaxed labeling itself. The algorithm does not operate with anything that can be interpreted as a relaxed labeling. In the following sections a stop criterium is defined as well as a method to compute the optimal relaxed labeling.

4 Stop condition of the subgradient descent algorithm

An evident test for power minimum is the consistency of condition (7) with respect to relaxed labeling (α, β). This test is too strict for subgradient descent. Realy, the condition (7) is consistent only at power's exact minimum. However, subgradient descent only converges to this minimum, it does not achieve the minimum itself in finite time. The test will be weakened by weakening both the condition (7) and the definition of the relaxed labeling (4).

Condition (7) is weakened by choosing some small positive number δ and replacing the condition with:

$$\begin{cases} q(t,k) < \max_{l \in K} q(t,l) - \delta & \Rightarrow \alpha(t,k) = 0 \; , \\ g((t,k),(t',k')) < \max_{l \in K, l' \in K} g((t,l),(t',l')) - \delta & \Rightarrow \beta((t,k),(t',k')) = 0 \; . \end{cases} \tag{12}$$

The existence of a relaxed labeling satisfying (12) does not guarantee the minimum, but indicates that the current power is close to minimum as well as the current relaxed labeling is close to the optimal as the following theorem states.

Theorem 1. *If the qualities (q, g) of a relaxed labeling problem with the power P are such that there exists a relaxed labeling (α, β) fulfilling the conditions (12) for a given δ, then*

$$P \le P^* + \delta \cdot (|T| + |\mathcal{T}|) \; ,$$

$$G(\alpha, \beta) \ge G(\alpha^*, \beta^*) - \delta \cdot (|T| + |\mathcal{T}|) \; ,$$

where P^* is the power minimum over all equivalent transformations and (α^*, β^*) is the optimal relaxed labeling.

The consistency of (12) w.r.t. (α, β) still gives some guaranties on the power value and relaxed labeling quality obtained. But as opposed to the original condition (7), the new one is achievable as the following theorem states.

Theorem 2. *Let the sequence $(q^i, g^i), i = 1, 2, \ldots$ be qualities of equivalent problems obtained by subgradient method. Then for any positive δ there exists such m that for qualities (q^m, g^m) the condition (12) is consistent w.r.t. (α, β).*

Therefore the subgradient method will eventually ensure the consistency of (12) w.r.t. (α, β).

Let us weaken the definition (4) of a relaxed labeling by introducing a small number ε and replacing the restriction (4) with

$$\begin{cases} \sum_{t \in T} \sum_{t' \in N(t)} \sum_{k \in K} \left[\alpha(t,k) - \sum_{k' \in K} \beta\big((t,k),(t',k')\big) \right]^2 \le \varepsilon \; , \\ \sum_{k \in K} \alpha(t,k) = 1, \qquad t \in T \; , \\ \alpha(t,k) \ge 0, \qquad t \in T, k \in K \; , \\ \beta\big((t,k),(t',k')\big) \ge 0, \quad tt' \in \mathcal{T}, k \in K, k' \in K \; . \end{cases} \tag{13}$$

With $\varepsilon = 0$ both (4) and (13) are equivalent. With any positive ε the found weights (α, β) may of course not be a relaxed labeling at all, but in some sense they are close to being a relaxed labeling and can be an acceptable substitute.

For each iteration i and obtained qualities (q^i, g^i) let us define the number S^i and call it the slack of the problem:

$$\begin{cases} S^i = \min \sum_{t\in T} \sum_{t'\in N(t)} \sum_{k\in K} \left[\alpha(t,k) - \sum_{k'\in K} \beta\big((t,k),(t',k')\big) \right]^2 , \\ \sum_{k\in K} \alpha(t,k) = 1, \qquad t \in T , \\ \alpha(t,k) \geq 0, \qquad t \in T, k \in K , \\ \beta\big((t,k),(t',k')\big) \geq 0, \quad tt' \in \mathcal{T}, k \in K, k' \in K , \\ q^i(t,k) < \max_{l\in K} q^i(t,l) - \delta \Rightarrow \qquad \alpha(t,k) = 0 , \\ g^i((t,k),(t',k')) < \max_{l\in K, l'\in K} g^i((t,l),(t',l')) - \delta \Rightarrow \quad \beta((t,k),(t',k')) = 0 . \end{cases} \tag{14}$$

Therefore, the stop condition for the subgradient descent can be defined as: if for the current qualities (q^i, g^i) the slack value S^i is greater than ε, then continue with subgradient minimization, if the slack value is less than ε, then the current problem is close to trivial and any relaxed labeling satisfying (12) is close to the optimal relaxed labeling.

Finding the exact solution to problem (14) is comparable to the original power minimization problem in terms of size and complexity. However, it is not necessary to find the exact value S^i to compare it with ε, it is sufficient to dispose appropriate upper and lower bound of the slack. If some upper bound of the slack is less than ε, then the subgradient descent should be stopped. Similarly, if some lower bound of the slack is bigger than ε, then the subgradient descent has to continue. The next sections describe how to find these lower and upper bounds.

5 Finding the upper and lower bound of the problem's slack

In this section the lower and upper bounds of the slack value are defined as functions of (α, β). First of all, let us introduce some notation to shorten the further used expressions:

$$\alpha \in A \Leftrightarrow \begin{cases} \sum_{k\in K} \alpha(t,k) = 1, \qquad t \in T , \\ \alpha(t,k) \geq 0, \qquad t \in T, k \in K , \\ q(t,k) < \max_{l\in K} q(t,l) - \delta \Rightarrow \quad \alpha(t,k) = 0 . \end{cases} \tag{15}$$

$$\beta \in B \Leftrightarrow \begin{cases} \beta\big((t,k),(t',k')\big) \geq 0, tt' \in \mathcal{T}, k \in K, k' \in K \ , \\ g((t,k),(t',k')) < \max\limits_{l \in K, l' \in K} g((t,l),(t',l')) - \delta \Rightarrow \\ \qquad \Rightarrow \beta((t,k),(t',k')) = 0 \ . \end{cases} \tag{16}$$

$$\Delta_{tt'}(k)(\alpha,\beta) = \alpha(t,k) - \sum_{k' \in K} \beta\big((t,k),(t',k')\big) \ . \tag{17}$$

Now the problem of finding S (13) can be expressed in terms of A, B, Δ:

$$S = \min_{\alpha \in A, \beta \in B} \sum_{t \in T} \sum_{t' \in N(t)} \sum_{k \in K} \Big[\Delta_{tt'}(k)(\alpha,\beta)\Big]^2 \ . \tag{18}$$

Suppose there is some approximate solution (α, β) that satisfies the conditions $\alpha \in A$, $\beta \in B$. Naturally, the upper bound $up(\alpha,\beta)$ of the problem can be defined as

$$up(\alpha,\beta) = \sum_{t \in T} \sum_{t' \in N(t)} \sum_{k \in K} \Big[\Delta_{tt'}(k)(\alpha,\beta)\Big]^2 \ . \tag{19}$$

Suppose the current weights (α,β) are such that $\beta = \arg\min\limits_{\beta' \in B} up(\alpha,\beta')$. Then for such weights the lower bound is defined:

$$\begin{aligned} low(\alpha,\beta) = & \frac{1}{up(\alpha,\beta)} \times \\ & \times \Bigg[\sum_{t \in T} \min_{k \in K_M(t)} \sum_{t' \in N(t)} \Delta_{tt'}(k)(\alpha,\beta) - \\ & \sum_{tt' \in \mathcal{T}} \sum_{k \in K} \sum_{k' \in K} \beta((t,k),(t',k')) \cdot \Big[\Delta_{tt'}(k)(\alpha,\beta) + \Delta_{t't}(k')(\alpha,\beta)\Big]\Bigg]^2 , \\ & k \in K_M(t) \Leftrightarrow g((t,k),(t',k')) \geq \max_{l \in K, l' \in K} g((t,l),(t',l')) - \delta \ . \end{aligned} \tag{20}$$

The $up(\alpha,\beta)$ and $low(\alpha,\beta)$ are in fact the upper and lower bound of S.

Theorem 3. *For any $\alpha \in A, \beta \in B$*

$$up(\alpha,\beta) \geq S \geq low(\alpha,\beta). \tag{21}$$

Theorem 4. *If the sequence of weights (α^i, β^i) satisfies*

$$\begin{cases} \alpha^i \in A \ , \\ \beta^i = \arg\min\limits_{\beta \in B} up(\alpha^i, \beta) \ , \\ \lim\limits_{i \to \infty} up(\alpha^i, \beta^i) = S \ , \end{cases} \tag{22}$$

then $\lim\limits_{i \to \infty} low(\alpha^i, \beta^i) = S$.

If the algorithm finds such (α,β) that $low(\alpha,\beta) > \varepsilon$, it is guaranteed that with the obtained qualities the conditions (13) and (12) are not consistent w.r.t. (α,β) and the subgradient descent has to continue.

6 Minimizing the upper bound

The algorithm for minimizing the upper bound(and at the same time maximizing the lower bound) consists of two main procedures. The first one minimizes the upper bound by α with fixed β and the other one minimizes the upper bound by β with fixed α. It is easy to see that by fixing one group of the variables (either α or β) the minimization by other group brakes down into smaller separate minimization problems which are much easier to solve than the original problem. Indeed, the restrictions $A = \bigotimes\limits_{t\in T} A_t$ in one pixel do not depend on any other pixel, and the restrictions $B = \bigotimes\limits_{tt'\in\mathcal{T}} B_{tt'}$ in one neighbouring pair do not depend on the other pair.

$$\min_{\alpha\in A} up(\alpha,\beta) = \min_{\alpha\in A}\sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\left[\alpha(t,k) - \sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right]^2 = \tag{23}$$

$$\sum_{t\in T}\min_{\alpha(t)\in A_t}\sum_{t'\in N(t)}\sum_{k\in K}\left[\alpha(t,k) - \sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right]^2 . \tag{24}$$

$$\min_{\beta\in B} up(\alpha,\beta) = \min_{\beta\in B}\sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\left[\alpha(t,k) - \sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right]^2 = \tag{25}$$

$$= \sum_{tt'\in\mathcal{T}}\min_{\beta(tt')\in B_{tt'}}\left[\sum_{k\in K}\left(\alpha(t,k) - \sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right)^2 + \sum_{k'\in K}\left(\alpha(t',k') - \sum_{k\in K}\beta\big((t,k),(t',k')\big)\right)^2\right]. \tag{26}$$

These separate minimizations are performed over a small set of variables, and so the exact minimum values can be found in finite time.

The whole algorithm for determining if sugradient descent has to stop looks like this:

1. *assign* $i := 0$
2. *initiate the weight values* (α^0,β^0) *with some numbers that fulfil* $\alpha^0 \in A, \beta^0 \in B$
3. *perform upper bound minimization by* α*:* $\alpha^{i+1} := \arg\min\limits_{\alpha\in A} up(\alpha,\beta^i)$
4. *perform upper bound minimization by* β*:* $\beta^{i+1} := \arg\min\limits_{\beta\in B} up(\alpha^{i+1},\beta)$

5. *calculate the* $up(\alpha^{i+1}, \beta^{i+1})$ *and* $low(\alpha^{i+1}, \beta^{i+1})$ *respectively*
6. *if* $up < \varepsilon$ *then stop subgradient descent and present* $(\alpha^{i+1}, \beta^{i+1})$ *as the relaxed labeling. Exit*
7. *if* $low > \frac{\varepsilon}{2}$ *then continue subgradient descent. Exit*
8. *Go to 3*

Theorem 5. *The algorithm stops in finite time.*

7 Experiments

The main idea of the proposed method is that the subgradient descent performs power minimization of any task while most known methods do not. There are known examples where the known methods do not ensure power minimization and one of them is presented in [3]. Subgradient descent method with proposed stop condition was tested on this example and its positive properties were validated. The goal of subsequent experiments was to determine how often it occurs in examples which are close to image processing.

In the following experiments the proposed stop condition was tested along with the well-known stop condition used in several algorithms (for example [1] [2]). The known stop condition also selects the set of edges and vertices that are close to maximum ones but instead of trying to build a relaxed labeling on it, checks for its consistency (see for example [2]). The same set of edges and vertices picked for both stop conditions. It is obvious that with $\varepsilon = 0$ the proposed stop condition is more strict than known one. With positive ε it may not be the case.

The set of pixels T forms a square field with the size $n \times n$. Each pixel (except for the ones on the borders of the field) has 4 neighbours: on the left, right, up and down.

To visualize the results the strict labeling is estimated from the obtained relaxed labeling by picking the label with the maximum weight in each pixel.

7.1 Vertical and horizontal lines

An image consists of black vertical and horizontal lines on a white background. It gets distorted by altering some pixel colors. The task is to restore the original image from the distorted one. More precisely, it is necessary to find a set of vertical and horizontal lines that produce an image such that the number of pixels different from the presented image is minimal.

The set of labels for this problem has 4 elements $VH, \overline{V}H, V\overline{H}, \overline{VH}$ with the following meaning: VH - both vertical and horizontal lines pass through the pixel. $\overline{V}H$ - only the horizontal line passes through the pixel. $V\overline{H}$ - only the vertical line passes through. $\overline{VH}$ - no lines pass through the pixel.

For two horizontal neighbours the edge qualities are picked such that the presence of a horizontal line must be the same in both pixels. For two vertical neighbours the edge qualities are picked such that the presence of a vertical line must be the same. These qualities of two horizontal neighbouring pixels are

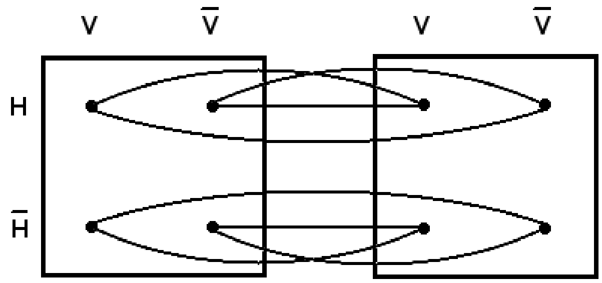


Fig. 1. Drawn edges have weight 0, others have $-\infty$.

shown on fig.1. The vertex qualities are produced from the distorted image. The black pixel has a (-1) weight for the label $\overline{VH}$ and 0 for the rest. The white pixel has a 0 weight for the label $\overline{VH}$ and (-1) for the rest.

Each line in table 1 describes one experiment with size, parameters δ and ε and power levels obtained by the known stop condition and the proposed one. In each experiment subgradient descent with the proposed stop condition has reached the power minimum. The last experiment from the table 1 is presented on fig. 2.

Table 1. Vertical and horizontal lines tests

Size	δ	ε	Consistency stop condition	Proposed stop condition
15×15	0.01	0.01	-35.911	-36
15×15	0.01	0.01	-28.973	-29
20×20	0.01	0.01	-49.986	-50
20×20	0.01	0.01	-60.971	-61
20×20	0.01	0.01	-55.975	-56

The proposed stop condition is slightly stricter than the known one in these examples.

7.2 Segmentation

The neighbouring structure is the same as with the first experiment. There are 3 labels, each corresponding to one color. For any neighbouring pixel pair (t, t') the weight function g is defined like this:

$$g_{tt'}(k, k') = \begin{cases} C, & k = k' \ , \\ 0, & k \neq k' \ . \end{cases} \tag{27}$$

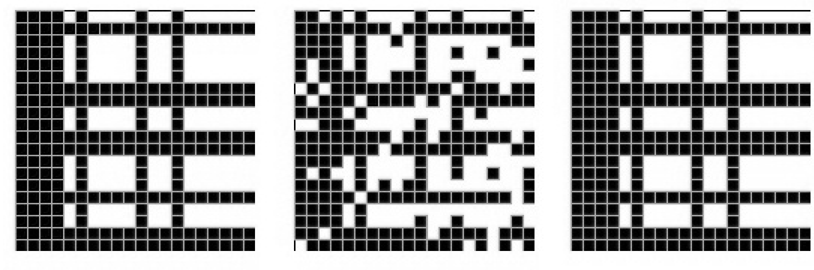


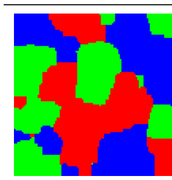
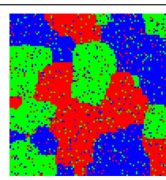
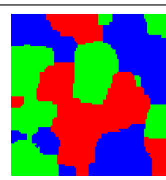
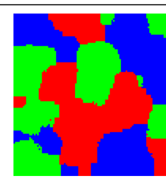
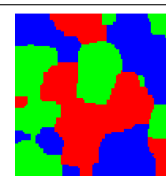
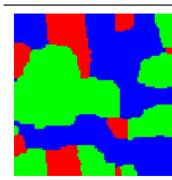
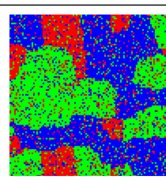
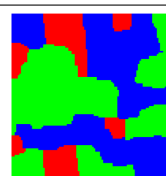
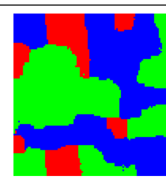
Fig. 2. From the left: original image, distorted image, restored image by the proposed stop condition.

In the following experiments $C = 3$.

Using these g and setting $q_t(k) = 0$ the images were generated with Gibbs sampler. Than each pixel has independently changed its color with some fixed probability p producing the distorted image. The new values q were obtained from the distorted image.

$$q_t(k) = \begin{cases} \ln(1 - p + \frac{p}{n}), & \text{k is the same as on the image} \\ \ln \frac{p}{n}, & \mathit{otherwise}\ . \end{cases} \tag{28}$$

Table 2. Segmentation

Original	Distorted	Consistency	$\varepsilon = 400$	$\varepsilon = 40$
				
100×100	10% noize	53311.1	53331.3	53311.7
				
100×100	20% noize	50794.9	50818.1	50804.9

The results are presented in the table 2. The five presented columns mean the following from left to right: the generated image with its size, the distorted image with the probability of a pixel to change its color, the image produced after known consistency-based stop condition was fulfilled with the power level obtained, the images produced with the proposed stop condition with ε value 400 and 40 with their respective power levels.

As one can see, in these examples the known stop condition showed itself to be more strict than the proposed one.

8 Concluding remarks

The experiments showed that the proposed stop condition can be used to determine power optimum as well as for estimating the optimal relaxed labeling. With some ε values the proposed stop condition is more strict than the known condition, with others not. In our experiments both stop conditions were approximately the same in terms of the power level obtained. The known stop condition has of course the advantage of being easier to compute. Moreover, there exist simpler and in some cases faster algorithms that achieve the known stop condition. On the other hand the advantage of the proposed method is that once it stops it simultaneously produces a relaxed labeling.

Thus both proposed and known stop conditions supplement each other, and it is reasonable to use them simultaneously in the following technological way.

1. The known stop condition is used together with algorithms that ensure the achievement of this condition, e.g., diffusion.
2. After the known stop condition is satisfied the proposed condition is checked.
3. If the proposed stop condition is satisfied the relaxed labeling is produced with the required precision.
4. If the proposed stop condition is not satisfied more complex algorithms have to be used that guarantee the achievement of the proposed condition, e.g., subgradient descent.

References

1. V. Kolmogorov. Convergent Tree-reweighted Message Passing for Energy Minimization. *In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568-1583, October 2006.
2. Schlesinger M.I., Antoniuk K.V., Diffusion algorithms and structural recognition optimization problems Cybernetics and Systems Analysis, 2011, Vol.48, Number 2, p.3-12 (In Russian). English translation of the paper will be published at Cybernetics and Systems Analysis, 2011, Number 2, Springer Verlag.
3. Schlesinger M.I., Giginjak V.V., Solving (max,+) problems of structural pattern recognition using equivalent transformations, Upravlyayushchie Sistemy i Mashiny (Control Systems and Machines), Kiev, Naukova Dumka, Number. 1 and 2, 2007, in Russian, English version is available at http://www.irtc.org.ua/image/publications/SchlGig_1_ru (Part 1), http://www.irtc.org.ua/image/publications/SchlGig_2_ru (Part 2).
4. Schlesinger M.I., Antoniuk K.V., Vodolazskii E.V. Optimal labeling problems, their relaxation and equivalent transformations.

5. Shor N. Nondifferentiable optimization and polynomial problems. - Dordrecht, Netherlannds: Kluwer Academic Publisher, 1998. - 394 p.
6. M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*, 51(11):36973717, November 2005.

A Appendix

Proof (Theorem 1).

$$\begin{aligned}
P^* \geq\ & G(\alpha,\beta) = \\
=\ & \sum_{t\in T}\sum_{k\in K}\alpha(t,k)\cdot q(t,k) + \sum_{tt'\in\mathcal{T}}\sum_{k\in K}\sum_{k'\in K}\beta((t,k),(t',k'))\cdot g((t,k),(t',k')) \geq \\
\geq\ & \sum_{t\in T}\sum_{k\in K}\alpha(t,k)\cdot\left[\max_{l\in K} q(t,l) - \delta\right] + \\
& + \sum_{tt'\in\mathcal{T}}\sum_{k\in K}\sum_{k'\in K}\beta((t,k),(t',k'))\cdot\left[\max_{l\in K,l'\in K} g((t,l),(t',l')) - \delta\right] = \\
=\ & \sum_{t\in T}\max_{l\in K} q(t,l) + \sum_{tt'\in\mathcal{T}}\max_{l\in K,l'\in K} g((t,l),(t',l')) - \delta\cdot|T| - \delta\cdot|\mathcal{T}| = \\
=\ & P - \delta\cdot(|T|+|\mathcal{T}|) \geq G(\alpha^*,\beta^*) - \delta\cdot(|T|+|\mathcal{T}|)\ .
\end{aligned} \tag{29}$$

So, $P^* \geq G(\alpha,\beta) \geq P - \delta\cdot(|T|+|\mathcal{T}|) \geq G(\alpha^*,\beta^*) - \delta\cdot(|T|+|\mathcal{T}|)$.

Proof (Theorem 2). Let P^* be the power minimum over all equivalent transformations. Let Θ be the set of all equivalent problems where the power function P reaches its minimum P^*. Let $D((q',g'),(q'',g''))$ be an Euclidean distance between two qualities and $D((q^i,g^i),\Theta)$ be the distance between a quality (q^i,g^i) and a set Θ. From subgradient descent theory it is known that

$$\lim_{i\to\infty} P(q^i,g^i) = P^*\ ,\ \lim_{i\to\infty} D((q^i,g^i),\Theta) = 0\ . \tag{30}$$

This means that for any positive δ there exists such k and such qualities $(q^*,g^*)\in\Theta$ that $D((q^k,g^k),(q^*,g^*)) < \frac{\delta}{2}$. Since the distance between two qualities is less than $\frac{\delta}{2}$, the qualities of each edge and each vertex also differ no more than $\frac{\delta}{2}$. This means that if two edges had the same quality in the problem (q^*,g^*), they would not differ by more than δ in the problem (q^k,g^k).

Since (q^*,g^*) is the optimum, there exits such relaxed labeling (α,β) that satisfy (7). And this very same (α,β) and qualities (q^k,g^k) also satisfy (12).

Proof (Theorem 3). The first inequality in (21) is trivial since $S = \min_{\alpha\in A,\beta\in B} up(\alpha,\beta)$. As for the second inequality the following is true:

$$\sqrt{up(\alpha,\beta)\cdot S} = \sqrt{up(\alpha,\beta)\cdot\min_{\alpha'\in A,\beta'\in B} up(\alpha',\beta')} = \tag{31}$$

$$= \sqrt{up(\alpha,\beta)\cdot \min_{\alpha'\in A,\beta'\in B}\sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\left[\alpha'(t,k)-\sum_{k'\in K}\beta'\big((t,k),(t',k')\big)\right]^2} \geq \tag{32}$$

$$\geq \min_{\alpha'\in A,\beta'\in B}\sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\left[\alpha'(t,k)-\sum_{k'\in K}\beta'\big((t,k),(t',k')\big)\right]\times \times\left[\alpha(t,k)-\sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right] = \tag{33}$$

$$= \min_{\alpha'\in A}\sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\alpha'(t,k)\times\left[\alpha(t,k)-\sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right] - \min_{\beta'\in B}\sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\sum_{k'\in K}\beta'\big((t,k),(t',k')\big)\times \times\left[\alpha(t,k)-\sum_{k'\in K}\beta\big((t,k),(t',k')\big)\right] = \tag{34}$$

$$= \sum_{t\in T}\min_{k\in k_M(t)}\sum_{t'\in N(t)}\Big(\alpha(t,k)-\sum_{k'}\beta((t,k),(t',k'))\Big) - \sum_{t\in T}\sum_{t'\in N(t)}\sum_{k\in K}\sum_{k'\in K}\beta((t,k),(t',k'))\cdot\Big(\alpha(t,k)-\sum_{k'}\beta((t,k),(t',k'))\Big)\ . \tag{35}$$

In 32 *up* is simply substituted with its definition. Inequality between 32 and 33 comes from the fact that the length of a vector is always not less than its projection. And 33 can be seen as projecting the vector (α',β') onto the vector (α,β). The equality of 33 and 34 is simply separating independent variables in the minimization. The first part of the last equality comes from the fact that the sum of weights α in one pixel is equal to 1 and they are positive numbers. The second part comes from β being the exact minimum of the upper bound with currently fixed α.

Proof (Theorem 4). By substituting the $up(\alpha,\beta)$ and $\Delta_{t't}(k')(\alpha,\beta)$ in (20) according to their definitions it can be shown that if for some (α^*,β^*), $up(\alpha^*,\beta^*) = S$ then $low(\alpha^*,\beta^*) = S$. Since both functions *low* and *up* are continuous over (α,β), $\lim_{i\to\infty} low(\alpha^i,\beta^i) = S$.

Proof (Theorem 5). The upper bound function $up(\alpha,\beta)$ is convex and continuously differentiable. The restrictions on the variables groups α and β are orthogonal. Therefore sequential minimizations converges to absolute minimum and either clause 6 or 7 of the algorithm will succeed.