Graph Cuts for Minimizing Robust Higher Order Potentials

Pushmeet Kohli
Microsoft Research Cambridge
pkohli@microsoft.com

L'ubor Ladický Philip H. S. Torr
Oxford Brookes University
{lladicky,philiptorr}@brookes.ac.uk

Abstract

Energy functions defined on higher order cliques can model complex interactions between groups of random variables. They have the capability of modelling the rich statistics of natural scenes which can be used for many applications in computer vision. However, these energies are seldom used in practice, primarily due to the lack of efficient algorithms for minimizing them. In this paper we introduce a novel family of higher order potentials called the Robust P^n model. This family is a generalization of the P^n Potts model class of potentials which was recently introduced in computer vision. We show that energy functions containing such potentials can be solved using the expansion and swap move algorithms for approximate energy minimization. Specifically, we prove that the optimal swap and expansion moves for energy functions composed of these potentials can be computed by solving a st-mincut problem. For functions of binary variables, our method is able to produce the globally optimal solution in polynomial time. Further, it is extremely efficient and can handle energies defined on cliques involving thousand of variables.

1. Introduction

Higher order potential functions can model complex interactions among sets of random variables. This flexibility enables them to encode sophisticated statistics of natural scenes which cannot be expressed using pairwise potentials. Researchers have used higher order potentials to produce excellent results for a number of challenging Computer Vision problems such as image restoration [18], denoising [12], optical flow [19], texture synthesis [16], and segmentation [7]. However, the lack of efficient algorithms for performing inference in these models has limited their popularity. The runtime complexity of commonly used inference algorithms such as belief propagation (BP) or Tree-reweighted message passing (TRW) [23] grows exponentially with the clique size, which makes them inapplicable to functions defined on even moderate sized cliques [12].

Recent work on message passing algorithms has been partly successful in improving their performance for certain classes of higher order potential functions. Lan *et al.* [12] proposed approximation methods for BP to make efficient inference possible in higher order MRFs. This was followed by the work of Potetz [17] in which he showed how belief propagation can be efficiently performed in graphical models containing moderately large cliques. However, as these methods are based on BP, they are quite slow and take minutes or even hours to converge. Kohli *et al.* [7] recently showed how certain higher order clique potentials can be minimized using the α-expansion and $\alpha\beta$-swap move making algorithms for approximate energy minimization [2]. They introduced a class of higher order potentials called the P^n Potts model and proved that the optimal *expansion* and *swap* moves for energy functions containing these potentials can be computed in polynomial time by solving a st-mincut problem.

In this paper we introduce a new family of higher order potentials called the Robust P^n model which is a generalization of the P^n Potts class. The potential functions belonging to this family are parameterized with a truncation parameter Q which controls their *robustness*. We will show how energy functions composed of these *robust* potentials can be minimized using move making algorithms such as α-expansion and $\alpha\beta$-swap. Specifically, we show how the optimal swap and expansion moves for such potentials can be found by solving an st-mincut problem. In fact, for functions of binary variables, our method produces the globally optimal solution. The complexity of our method increases linearly with the clique size and thus it is able to handle cliques composed of thousands of latent variables. In a related paper on object segmentation, we have shown that the Robust P^n model potentials produce more accurate segmentation results compared to those obtained using pairwise and/or P^n Potts potential functions [8].

Outline of the Paper A brief outline of the paper follows. In section 2 we discuss the energy minimization problem and describe the graph cut based expansion and swap move algorithms. In section 3 we introduce the Robust P^n model and discuss its relationship to the P^n Potts model. We also describe how it can be used to approximate any concave increasing consistency potential function. In section

4, we show how the optimal swap and expansion moves for functions containing the Robust P^n model potentials can be computed by solving an st-mincut problem. The experimental results are provided in section 5. The proofs for the theorems stated in the paper are given in the appendix.

2. Energy Minimization

Many problems in computer vision and artificial intelligence can be formulated in terms of minimizing an energy function $E(\mathbf{x})$. This function is defined on a set of discrete random variables $\mathbf{X} = \{X_1, X_2, \ldots, N\}$ each of which takes a value from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$. A labelling or *configuration* $\mathbf{x} \in \mathcal{L}^N$ is a possible assignment of labels to the random variables.

The energy function associated with a labelling problem is defined in terms of the posterior probability distribution of the possible configurations of the random variables, i.e. $E(\mathbf{x}) = -\log \Pr(\mathbf{x}|\mathbf{D}) - \log Z$, where Z is the partition function. Further, they can be written in terms of a sum of potential functions defined over cliques of variables as: $E(\mathbf{x}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c)$, where $\psi_c(\mathbf{x}_c)$ is the potential function defined over the clique c where $\mathbf{x}_c = \{x_i, i \in c\}$. Given the above notation, the maximum a posterior (MAP) labelling $\mathbf{x_{map}}$ of the random variables can be found as

$$\mathbf{x_{map}} = \arg\max_{\mathbf{x} \in \mathbf{L}} \Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x} \in \mathbf{L}} E(\mathbf{x}). \quad (1)$$

The energy minimization problem (1) is NP-hard in general [9]. However, there exist classes of functions which can be minimized exactly in polynomial time. Two well known classes of tractable functions are: sub-modular functions, and functions defined over graphs with tree topology. However, most energies encountered in practical problems do not belong to these families. They are instead solved using algorithms for approximate energy minimization. These algorithms can be divided into two broad categories: message passing algorithms such as belief propagation and its variants [26, 23, 9], and move making algorithms such as the graph cut based α-expansion and $\alpha\beta$-swap [2].

Message passing algorithms have been shown to produce excellent results for many energy functions. However, their runtime complexity increases exponentially with the size of the largest clique making them inapplicable to functions defined over large cliques. In this paper we show how such energy functions can be minimized efficiently using the graph cut based α-expansion and $\alpha\beta$-swap move algorithms.

2.1. Expansion and Swap Move Algorithms

Move making algorithms start from an initial solution and proceed by making a series of changes which lead to solutions having lower energy. At each step, the algorithms search a move space and choose the move which leads to the solution having the lowest energy. This move is referred to as the *optimal* move. The algorithm is said to converge when no lower energy solution can be found.

The size of the move space is a key characteristic of these algorithm. A large move space means that bigger changes to the current solution can be made. This makes the algorithm less prone to getting stuck in local minima and also results in a faster rate of convergence. This paper deals with two particular *large* move making algorithms, namely α-*expansion* and $\alpha\beta$-*swap* [2] whose move space size increases exponentially with the number of variables involved in the energy function. We will use the notation of [7] to describe how these algorithms work. The moves of the expansion and swap algorithms can be encoded as a vector of binary variables $\mathbf{t} = \{t_i, \forall i \in \mathcal{V}\}$. The *transformation* function $T(\mathbf{x}^p, \mathbf{t})$ of a move algorithm takes the current labelling $\mathbf{x}^p$ and a move $\mathbf{t}$ and returns the new labelling $\mathbf{x}^n$ which has been induced by the move.

An α-expansion move allows any random variable to either retain its current label or take label 'α'. One iteration of the algorithm involves making moves for all α in $\mathcal{L}$ in some order successively. The transformation function $T_\alpha()$ for an α-expansion move transforms the label of a random variable X_i as

$$T_\alpha(x_i, t_i) = \begin{cases} \alpha & \text{if} \quad t_i = 0 \\ x_i^p & \text{if} \quad t_i = 1. \end{cases} \quad (2)$$

An $\alpha\beta$-swap move allows a random variable whose current label is α or β to either take label α or β. One iteration of the algorithm involves performing swap moves for all α and β in $\mathcal{L}$ in some order successively. The transformation function $T_{\alpha\beta}()$ for an $\alpha\beta$-swap transforms the label of a random variable x_i as

$$T_{\alpha\beta}(x_i, t_i) = \begin{cases} \alpha & \text{if} \quad x_i = \alpha \text{ or } \beta \text{ and } t_i = 0, \\ \beta & \text{if} \quad x_i = \alpha \text{ or } \beta \text{ and } t_i = 1. \end{cases} \quad (3)$$

The energy of a move $\mathbf{t}$ is the energy of the labelling $\mathbf{x}$ the move $\mathbf{t}$ induces i.e. $E_m(\mathbf{t}) = E(T(\mathbf{x}, \mathbf{t}))$. The move energy will be denoted by $E_m(\mathbf{t})$. At each step of the expansion and swap move algorithms, the *optimal* move $\mathbf{t}^*$, i.e. the move decreasing the energy of the labelling by the most amount is computed. This is done by minimizing the move energy i.e. $\mathbf{t}^* = \arg\min_{\mathbf{t}} E(T(\mathbf{x}, \mathbf{t}))$. The optimal move $\mathbf{t}^*$ can be computed in polynomial time if the move function $E_m(\mathbf{t})$ is sub-modular.

2.2. Related Work

The last few years have seen a lot of interest in graph cut based move algorithms for energy minimization. Komodakis *et al.* [10, 11] recently gave an alternative interpretation of the α-expansion algorithm. They showed that

α-expansion works by solving the dual of a linear programming relaxation of the energy minimization problem. Using this theory, they developed a new algorithm (FAST-PD) which was faster than vanilla α-expansions and produced the exact same solution.

Researchers have also proposed a number of novel move encoding strategies for solving particular forms of energy functions. For instance, Winn and Shotton [25] proposed a move algorithm in which variables are allowed to move to different labels in each iteration (rather than just one as in α-expansion). Veksler [22] proposed a move algorithm in which variables can choose any label from a range of labels. They showed that this move space allowed them to obtain better minima of energy functions with truncated convex pairwise terms. More recently, Lempitsky, Rother and Blake [13] proposed an algorithm which encoded labels by a binary number. During each move, the variables were allowed to change a particular bit of the binary variable. They showed that this particular move encoding strategy results in a substantial speedup when minimizing energy functions with large label sets.

2.3. Submodular functions

Submodular set functions are encountered in many areas of research and are particularly useful in combinatorial optimization, probability and geometry [5, 14]. Many optimization problems relating to submodular functions can be solved efficiently. In this respect they are similar to convex/concave functions encountered in continuous optimization. We will now give the formal definition of a submodular function. For this however, we will first need to define the concept of a projection of a function. A projection of a function $f : \mathcal{L}^n \rightarrow \mathbb{R}$ on s variables is a function $f^p : \mathcal{L}^s \rightarrow \mathbb{R}$ which is obtained by fixing the values of $n-s$ arguments of $f(\cdot)$. Here p refers to the set of variables whose values have been fixed. A function of one binary variable is always submodular. A function $f(x_1, x_2)$ of two binary variables $x_1, x_2 \in \mathbb{B} = \{0, 1\}$ is submodular if and only if: $f(0,0) + f(1,1) \leq f(0,1) + f(1,0)$. A function $f : \mathbb{B}^n \rightarrow R$ is submodular if and only if all its projections on 2 variables are submodular [1, 9].

2.4. Computing Moves using Graph Cuts

We had discussed in section 2.1 that the optimal expansion and swap moves can be computed by minimizing the move energy function. A move function can be minimized in polynomial time if it is submodular [15]. However, algorithms for submodular function minimization have high runtime complexity. Although recent work has been successful in reducing the runtime complexity of these algorithms, they are still quite computationally expensive and cannot be used to minimize large functions. For instance, the current best algorithm for general submodular function minimization has complexity $O(n^5\mathcal{T} + n^6)$ where $\mathcal{T}$ is the time taken to evaluate the function [15]. This algorithm improved upon the previous best strongly polynomial time algorithm by a factor of $n \log n$.

Certain submodular functions can be minimized by solving an st-mincut problem [1]. Specifically, all submodular functions of binary variables of order at most 3 can be minimized in this manner [9]. Researchers have shown that certain higher order functions can be transformed into submodular functions of order 2, and thus can also be minimized [4]. The same transformation technique can be used to minimize some functions of multi-valued variables [3, 6, 20]. Boykov *et al.* [2] showed that the optimal moves for certain pairwise energy functions can be computed by solving an st-mincut problem. Specifically, they showed that the optimal expansion move for a second-order energy function can be computed by solving an st-mincut problem if the pairwise potential functions define a metric. Similarly, the optimal $\alpha\beta$-swap move can be computed if the pairwise potentials defines a semi-metric.

3. Robust Higher Order Potentials

Kohli *et al.* [7] recently characterized a class of higher order clique potentials for which the optimal expansion and swap moves can be computed by minimizing a submodular function. However, as discussed earlier, minimizing a general submodular function is quite computationally expensive and thus it is infeasible to apply this procedure for minimizing energy functions encountered in computer vision problems which generally involve millions of random variables. In the same work, they also introduced a class of higher order clique potentials called the P^n Potts model and showed that the optimal *expansion* and *swap* moves for energy functions containing these potentials can be computed in polynomial time by solving a st-mincut problem. The P^n Potts model was defined as:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma_k & \text{if } x_i = l_k, \forall i \in c, \\ \gamma_{\max} & \text{otherwise.} \end{cases} \quad (4)$$

where $\gamma_{\max} \geq \gamma_k,\ \forall l_k \in \mathcal{L}$. This potential is a higher order generalization of the widely used Potts model potential which is defined over cliques of size two as $\psi_{ij}(a, b) = \gamma_k$ if $a = b = l_k$ and $\gamma_{\max}$ otherwise.

In this paper we introduce a novel family of higher order potentials which we call the Robust P^n model. This family contains the P^n Potts model as well as its *robust* variants, and can be used for modelling many computer vision problems. We show that the optimal swap and expansion move energy functions for any Robust P^n model potential can be transformed into a second order submodular function by the addition of at most two binary auxiliary variables. This transformation enables us to find the optimal swap and ex-

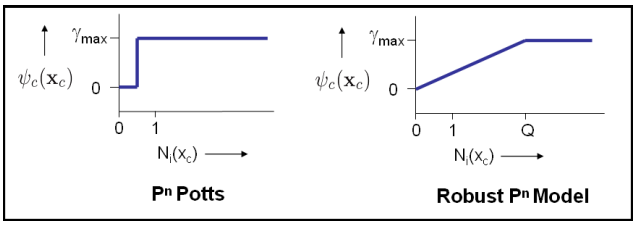


Figure 1. *Behaviour of the rigid P^n Potts potential and the Robust P^n model potential. The figure shows how the cost enforced by the two higher order potentials changes with the number of variables in the clique not taking the dominant label i.e. $N_i(\mathbf{x}_c) = min_k(|c| - n_k(\mathbf{x}_c))$.*

pansion moves in polynomial time[1].

The problem of transforming a general submodular higher order function to a second order one has been well studied. Kolmogorov and Zabih [9] showed that all submodular functions of order three can be transformed to one of order two, and thus can be solved using graph cuts. Freedman and Drineas [4] showed how certain submodular higher order functions can be transformed to submodular second order functions. However, their method in the worst case needed to add exponential number of auxiliary binary variables to make the energy function second order. Due to the special form of the Robust P^n model, the method described in the paper only needs to add two binary variables per higher order potential to make the move energy functions submodular. This allows for the efficient computation of the optimal moves. The complexity of our algorithm for computing the optimal move increases linearly with the size of the clique. This enables us to handle potential functions defined over very large cliques.

The Robust P^n model potentials take the form:

$$\psi_c(\mathbf{x}_c) = \min\{\min_{k\in\mathcal{L}}((|c| - n_k(\mathbf{x}_c))\theta_k + \gamma_k), \gamma_{\max}\} \quad (5)$$

where $|c|$ is the number of variables in clique c, $n_k(\mathbf{x}_c)$ denotes the number of variables in clique c which take the label k in labelling $\mathbf{x}_c$, and $\gamma_k, \theta_k, \gamma_{\max}$ are potential function parameters which satisfy the constraints:

$$\theta_k = \frac{\gamma_{\max} - \gamma_k}{Q} \text{ and } \gamma_k \leq \gamma_{\max}, \forall k \in \mathcal{L}. \quad (6)$$

Q is called the truncation parameter of the potential and satisfies the constraint $2Q < |c|$. It can be seen that the Robust P^n model (5) becomes a P^n Potts model (4) when the truncation parameter is set to 1.

Example 1. *Consider the set of variables $\mathbf{X} = \{X_1, X_2, \ldots, X_7\}$ where each $X_i, i \in \{1, 2, \ldots, 7\}$ takes a value from the label set $\mathcal{L} = \{a, b, c\}$. The P^n Potts model assigns the cost $\gamma_{\max}$ to all labellings of the random variables except those where all variables X_i take the same label. Thus, the configuration $\mathbf{x} = (a, a, b, a, c, a, a)$ will be assigned cost $\gamma_{\max}$ even though there are only 2 variables (specifically, X_3 and X_5) which are assigned labels (b and c) different from the dominant label a. In contrast, the Robust P^n model with truncation 3, i.e. $Q = 3$, assigns the cost: $\gamma_a + \frac{(\gamma_{\max} - \gamma_a)}{3} \times 2$ to the same configuration.*

[1] All second order submodular functions of binary variables can be minimized exactly in polynomial time by solving an st-mincut problem [1, 9].

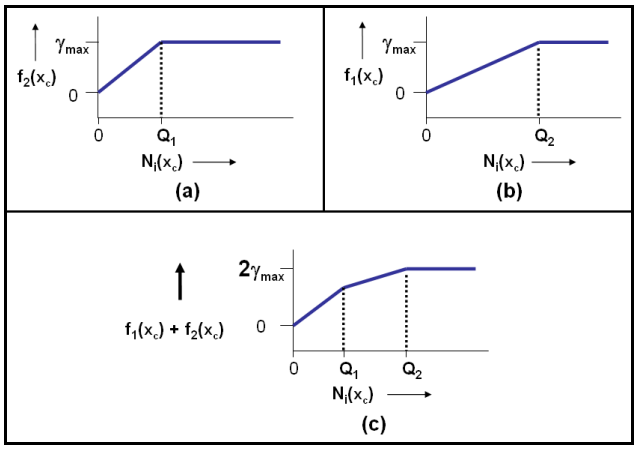


Figure 2. *Approximating Concave Consistency Potentials. The figure shows the result of combining two robust higher order potentials (a) and (b). The resulting potential function is shown in (c).*

3.1. Approximating Concave Consistency Potentials

Let us now consider the Robust P^n model where the constants γ_k have the same value for all labels $k \in \mathcal{L}$. In this case, the higher order potential can be seen as encouraging all the variables in the clique c to take the same label. In other words, the potential tries to reduce the number of variables in the clique not taking the dominant label i.e. $N_i(\mathbf{x}_c) = \min_k(|c| - n_k(\mathbf{x}_c))$. In what follows we will refer to these variables as *inconsistent*.

Unlike the P^n Potts model which enforces label consistency very rigidly, the Robust P^n Potts model gives rise to a cost that is a linear truncated function of the number of inconsistent variables (see figure 1). This enables the robust potential to allow some variables in the clique to take different labels. Multiple Robust P^n model potentials can be combined to approximate any non-decreasing concave consistency potential up to an arbitrary accuracy. These potentials take the form:

$$\psi_c(\mathbf{x}_c) = \min\{\min_{k\in\mathcal{L}} \mathcal{F}_c((|c| - n_k(\mathbf{x}_c))), \gamma_{\max}\} \quad (7)$$

where $\mathcal{F}_c$ is a non-decreasing concave function[2]. This is illustrated in figure 2.

[2] A function $f(x)$ is concave if for any two points (a, b) and λ where $0 \leq \lambda \leq 1$: $\lambda f(a) + (1 - \lambda)f(b) \leq f(\lambda a + (1 - \lambda)b)$.

3.2. Generalized form of Robust Higher Order Potentials

We now provide a characterization of a larger class of functions for which at most two auxiliary variables are sufficient to transform the higher order swap and expansion move energy functions to second order functions. The potentials belonging to this new family have the form:

$$\psi_c(\mathbf{x}_c) = \min\{\min_{k\in\mathcal{L}}((P - f_k(\mathbf{x}_c))\theta_k + \gamma_k), \gamma_{\max}\} \quad (8)$$

and $\gamma_k, \theta_k, \gamma_{\max}$ are potential function parameters which satisfy the constraints:

$$\theta_k = \frac{\gamma_{\max} - \gamma_k}{Q_k} \text{ and } \gamma_k \leq \gamma_{\max}, \forall k \in \mathcal{L}. \quad (9)$$

$Q_k, k \in \mathcal{L}$ are the truncation parameters of the potential functions and satisfy the constraints $Q_a + Q_b < P, \forall a \neq b \in \mathcal{L}$. The functions $f_k(\mathbf{x}_c)$ are defined as:

$$f_k(\mathbf{x}_c) = \sum_{i\in c} w_i^k \delta_k(x_i) \quad (10)$$

$$\text{where} \quad \delta_k(x_i) = \begin{cases} 1 & \text{if} \quad x_i = k, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

and weights $w_i^k \geq 0, i \in c, k \in \mathcal{L}$ encode the relative importance of different variables in preserving consistency of the labelling of the clique. They satisfy the constraint:

$$\sum_{i\in c} w_i^k = P, \quad \forall k \in \mathcal{L}. \quad (12)$$

If we assume that $w_i^k = w_i \geq 0$ and $Q_k = Q$ for all $k \in \mathcal{L}$, the potential family (8) can be seen as weighted version of the Robust P^n model. The weights can be used to specify the relative importance of different variables. This can be used in the 'segment merging' application described in [8] by reducing the inconsistency cost for pixels on the segment boundary by reducing their weights. We will show that for the case of symmetric weights i.e. $w_i^k = w_i$, the higher order swap and expansion and move energy functions for the potentials (8) can be transformed to a submodular second order binary energy.

4. Transforming Higher Order Move Energies

We will now explain how the optimal swap and expansion moves for energy functions containing potential functions of the form (8) can be computed using graph cuts. In what follows we show that any swap or expansion move energy for higher order potentials of the form (8) can be converted to a sub-modular pairwise function if $w_i^k = w_i$ for all $k \in \mathcal{L}$. To proceed further, we will need to define the function $W(s), s \subseteq c$:

$$W(s) = \sum_{i\in s} w_i. \quad (13)$$

It can be seen from constraint (12) that $W(c) = P$.

4.1. Swap Moves

Recall from the definition of the swap move transformation function that only variables which are currently assigned labels α or β can take part in a $\alpha\beta$-swap move. We call these variables *active* and denote the vector of their indices by c_a. $\mathbf{t}_{c_a}$ will be used to denote the corresponding vector of move variables. Similarly, variables in the clique which do not take part in the swap move are called *passive*, and the set of their indices is denoted by c_p. Let functions $f_k^m(\mathbf{t}_{c_a}), k \in \{0, 1\}$ be defined as:

$$f_k^m(\mathbf{t}_{c_a}) = \sum_{i\in c_a} w_i \delta_k(t_i). \quad (14)$$

The move energy of a $\alpha\beta$-swap move from the current labelling $\mathbf{x}_c^p$ is equal to the energy of the new labelling $\mathbf{x}_c^n$ induced by the move and is given as

$$\psi_c^m(\mathbf{t}_{c_a}) = \psi_c(\mathbf{x}_c^n). \quad (15)$$

The new labelling $\mathbf{x}_c^n$ is obtained by combining the old labelling of the passive variables $\mathbf{X}_{c_p}$ with the new labelling of the active variables $\mathbf{X}_{c_a}$ as:

$$\mathbf{x}_c^n = \mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}). \quad (16)$$

On substituting the value of $\mathbf{x}_c^n$ from (16) in (15), and using the definition of the higher order potential functions (8) we get:

$$\psi_c^m(\mathbf{t}_{c_a}) = \psi_c(\mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a})) \quad (17)$$

$$= \min\{\min_{k\in\mathcal{L}}(z_k\theta_k + \gamma_k), \gamma_{\max}\} \quad (18)$$

where $z_k = P - f_k(\mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}))$.

It can be easily observed that if conditions:

$$W(c_a) < P - Q_\alpha \quad \text{and} \quad W(c_a) < P - Q_\beta, \quad (19)$$

are satisfied, then the expression:

$$(P - f_k(\mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a})))\theta_k + \gamma_k \quad (20)$$

is greater than $\gamma_{\max}$ for both $k = \alpha$ and $k = \beta$. Thus, in this case the move energy $\psi_c^m(\mathbf{t}_{c_a})$ is independent of $\mathbf{t}_{c_a}$ and is equal to the constant:

$$\eta = \min\{\min_{k\in\mathcal{L}\setminus\{\alpha,\beta\}}((P - f_k(\mathbf{x}_c^p))\theta_k + \gamma_k), \gamma_{\max}\} \quad (21)$$

which can be ignored while computing the swap moves. However, if constraints (19) are not satisfied, the move energy becomes: $\psi_c^m(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}) =$

$$\min\{(W(c_a) - f_0^m(\mathbf{t}_{c_a}))\theta_\alpha + \lambda_\alpha, (W(c_a) - f_1^m(\mathbf{t}_{c_a}))\theta_\beta + \lambda_\beta, \lambda_{\max}\} \quad (22)$$

where $\lambda_\alpha = \gamma_\alpha + R_{\alpha\beta}\theta_\alpha$, $\lambda_\beta = \gamma_\beta + R_{\alpha\beta}\theta_\beta$, $\lambda_{\max} = \gamma_{\max}$ and $R_{\alpha\beta} = W(c - c_a)$.

The minimization in (22) can be removed by defining the move energy as:

$$\psi_c^m(\mathbf{t}_{c_a}) = \begin{cases} K_\alpha & \text{if} \quad f_0^m(\mathbf{t}_{c_a}) > W(c_a) - \hat{Q}_\alpha \\ K_\beta & \text{if} \quad f_0^m(\mathbf{t}_{c_a}) \leq \hat{Q}_\beta \\ \lambda_{\max} & \text{otherwise} \end{cases} \tag{23}$$

where $\hat{Q}_k = Q_k - R_{\alpha\beta},\ k \in \{\alpha, \beta\}$, $K_\alpha = \lambda_\alpha + (W(c_a) - f_0^m(\mathbf{t}_{c_a}))\theta_\alpha$, and $K_\beta = \lambda_\beta + f_0^m(\mathbf{t}_{c_a})\theta_\beta$.

Next we will show that this higher order move energy can be written as a second order submodular function with the addition of the auxiliary binary variables m_0 and m_1.

Theorem 1. *The higher order move energy (23) can be transformed into a pairwise energy function by introducing binary meta-variables m_0 and m_1 as:*

$$\begin{aligned} \psi_c^m(\mathbf{t}_c) \quad = \quad & \min_{m_0, m_1} r_0(1 - m_0) + \theta_\beta m_0 \sum_{i \in c_a} w_i(1 - t_i) \\ + \quad & r_1 m_1 + \theta_\alpha(1 - m_1) \sum_{i \in c_a} w_i t_i - \delta, \end{aligned} \tag{24}$$

where $r_0 = \lambda_\alpha + \delta$, $r_1 = \lambda_\beta + \delta$, and $\delta = \lambda_{\max} - \lambda_\alpha - \lambda_\beta$.

Proof. Proof is given in the appendix. □

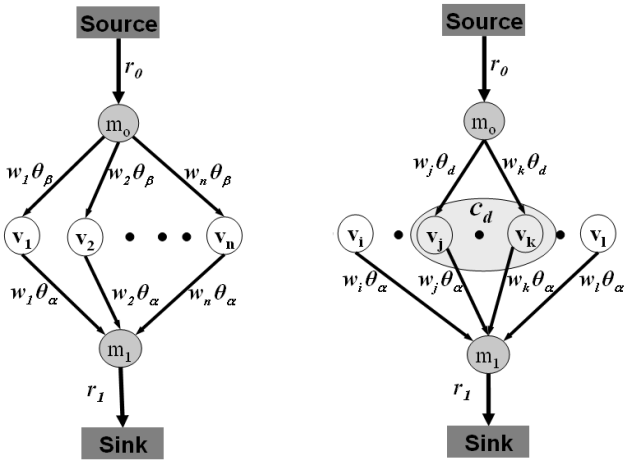
The properties $\gamma_{\max} \geq \gamma_k, \forall k \in \mathcal{L}$ and $w_i \geq 0$ of the clique potential (8) imply that all coefficients of the energy function (24) are non-negative. The function is thus *submodular* and can be minimized by solving a st-mincut problem [9]. The graph construction for minimizing the energy function (24) is shown in figure (3a). The constant δ in (24) does not affect the minimization problem i.e. it does not change the move having the least energy and thus is ignored.

Exact minimization of functions of binary variables For functions of binary variables, the swap move energy is the same as the original energy function, i.e. $\psi_c^m(\cdot) \equiv \psi_c(\cdot)$. Thus, the method described above can be used to perform exact minimization of such functions.

4.2. Expansion Moves

We now describe how the optimal expansion moves can be computed for the higher order potentials (8).

Let c_k denote the set of indices of variables in clique c that have been assigned label k in the current solution $\mathbf{x}_c^p$. We find the *dominant* label $d \in \mathcal{L}$ in $\mathbf{x}_c^p$ such that $W(c_d) > P - Q_d$ where $d \neq \alpha$. The constraints $Q_a + Q_b < P$, $\forall a \neq b \in \mathcal{L}$ of the higher order potentials (8) make sure that there is at most one such label. If we find such a label in the current labelling, then the expansion



(a) Swap Move (b) Expansion Move

Figure 3. *(a) Graph construction for minimizing the swap energy function (24). (b) Graph construction for minimizing the expansion move energy function (27). For every binary move variable t_i in the energy function there is a corresponding graph node v_i. The minimum cost source sink cut (st-mincut) divides the graph into two sets: the source set ($\mathcal{S}$) and the sink set ($\mathcal{T}$). $v_i \in (\mathcal{S})$ implies $t_i = 1$ while $v_i \in (\mathcal{T})$ implies $t_i = 0$.*

move energy can be written as: $\psi_c^m(\mathbf{t}_c) = \psi_c(T_\alpha(\mathbf{x}_c^p, \mathbf{t}_c))$ or, $\psi_c^m(\mathbf{t}_c) =$

$$\min\{\lambda_\alpha + \theta_\alpha \sum_{i \in c} w_i t_i, \lambda_d + \theta_d \sum_{i \in c_d} w_i(1 - t_i), \lambda_{\max}\} \tag{25}$$

where $\lambda_\alpha = \gamma_\alpha$, $\lambda_d = \gamma_d + R_d\theta_d$, $\lambda_{\max} = \gamma_{\max}$ and $R_d = W(c - c_d)$. Without the minimization operator the function (25) becomes:

$$\psi_c^m(\mathbf{t}_c, \mathbf{t}_{c_d}) = \begin{cases} K_\alpha & \text{if} \quad f_0^m(\mathbf{t}_c) > P - Q_\alpha \\ K_d & \text{if} \quad f_0^m(\mathbf{t}_{c_d}) \leq Q_d - R_d \\ \lambda_{\max} & \text{otherwise} \end{cases} \tag{26}$$

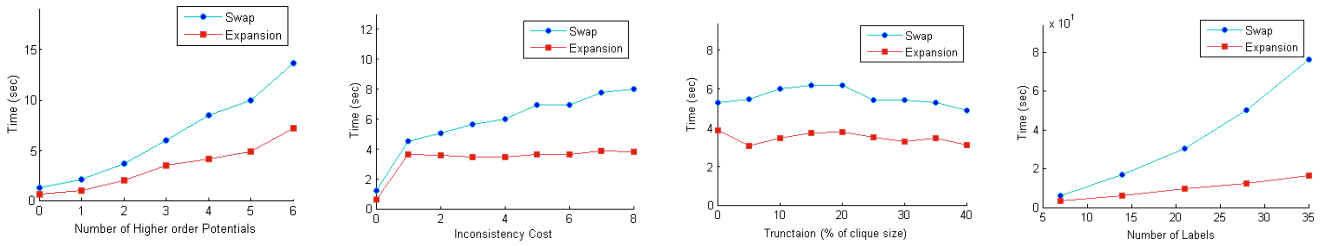
where $K_\alpha = \lambda_\alpha + (P - f_0^m(\mathbf{t}_c))\theta_\alpha$, and $K_d = \lambda_d + f_0^m(\mathbf{t}_{c_d})\theta_d$. Next we will show that this higher order move energy can be written as a second order submodular function with the addition of the auxiliary binary variables m_0 and m_1.

Theorem 2. *The expansion move energy (26) can be transformed into the pairwise function:*

$$\begin{aligned} \psi_c^m(\mathbf{t}_c) \quad = \quad & \min_{m_0, m_1} r_0(1 - m_0) + \theta_d m_0 \sum_{i \in c_d} w_i(1 - t_i) \\ + \quad & r_1 m_1 + \theta_\alpha(1 - m_1) \sum_{i \in c} w_i t_i - \delta. \end{aligned} \tag{27}$$

where $r_0 = \lambda_\alpha + \delta$, $r_1 = \lambda_d + \delta$, and $\delta = \lambda_{\max} - \lambda_\alpha - \lambda_d$.

Proof. Proof in appendix. □



(a) No. of Potentials (b) Inconsistency Cost (c) Truncation (d) No. of Labels

Figure 4. *Convergence times of the swap and expansion move algorithms. The graphs show how the convergence times of the move algorithms is affected by changes in the higher order energy function. The experiments were performed on grids corresponding to 10 randomly selected images from the Sowerby dataset for Object Segmentation. The size of the grid was* 96×64. *The unary and pairwise potentials were generated using the method proposed in [21]. Higher order potentials of the form of the Robust* P^n *were generated randomly and incorporated in the conditional random field. A detailed description of the graphs can be found in the text. The average convergence time (in seconds) of the expansion, swap and ICM algorithms (in this order) for the different experiments were: (a) 3.3, 6.4, 312.3, (b) 3.5, 5.6, 384.6, (c) 3.3, 5.8, 318.4 (d) 9.6, 35.9, 298.1.*

The energy function (27) is submodular and can be minimized by finding the st-mincut in the graph shown in figure (3b).

If a dominant label cannot be found then the move energy can be written as:

$$\psi_c^m(\mathbf{t}_c) = \min\{\lambda_\alpha + \theta_\alpha \sum_{i \in c} w_i t_i, \lambda_{\max}\} \qquad (28)$$

where $\lambda_\alpha = \gamma_\alpha$, and $\lambda_{\max} = \gamma_{\max}$. This can be transformed to the binary pairwise energy:

$$\psi_c^m(\mathbf{t}_c) = r_1 m_1 + \theta_\alpha (1 - m_1) \sum_{i \in c} w_i t_i + \lambda_\alpha, \qquad (29)$$

where $r_1 = \lambda_{\max} - \lambda_\alpha$. The proof for this transformation is similar to the one shown for Theorem 2.

5. Experiments

We now provide the results of our experiments on solving higher order energy functions containing Robust P^n model potentials using the methods proposed in this paper. In a companion paper, we show that incorporation of Robust P^n model potentials in the conditional random field model for object segmentation leads to a significant improvement in the segmentation results [8].

We have tested our methods on randomly generated higher order energy functions. We compare the performance of our methods with the Iterated Conditional Modes (ICM) algorithm. Comparison with the conventional factor graph formulation [12] of message passing algorithms like BP and TRW-S was infeasible due to the large size of cliques defining the energy functions used in our tests[3]. Our experiments show that the graph cut based expansion and swap move algorithms for the Robust P^n model potentials produce solutions with lower energy compared to the ICM algorithm. Further, they required much less time to converge to the final solution compared to ICM. The graph in fig. 5 show how the energy of the solutions obtained from different minimization algorithms changes with time. The graphs in fig. 4 show how the convergence time for the different algorithm is influenced by parameters of the energy function.

The graph in figure 4(a) shows how the convergence time is affected by the number of higher order potentials. In the energy functions used for this experiment, each random variable is included in the same number of higher order cliques. The x-axes of the graph shows the number of higher order potentials each variable is involved in. As expected the convergence time increases with the number of higher order potentials in the energy function. The graph in figure 4(b) shows the effect of parameter $\gamma_{\max}$ of the Robust P^n model potentials on the convergence time. The graph in figure 4(c) shows the effect of the truncation parameter Q of the Robust P^n model. Q is specified as the percentage of the size of the higher order clique. The change in convergence time of the move making algorithms with the increase in size of the label set of the random variables can be seen in graph 4(d).

6. Discussion and Conclusion

The importance of higher order potentials in computer vision cannot be overstated. Recent research in low-level vision has illustrated the usefulness of higher order image statistics [18, 24]. It is apparent that utilizing this information is essential for a proper treatment of these problems. To realize this, we need to develop efficient inference algo-

[3]It should be noted that the Robust P^n model potentials can be transformed into pairwise potentials by the addition of multi-label auxiliary variables. This enables the use of message passing algorithms for minimizing energy functions composed of them. However, the analysis of these transformations, and the subsequent study of the performance of message passing algorithms on the transformed functions lies outside the scope of this paper.

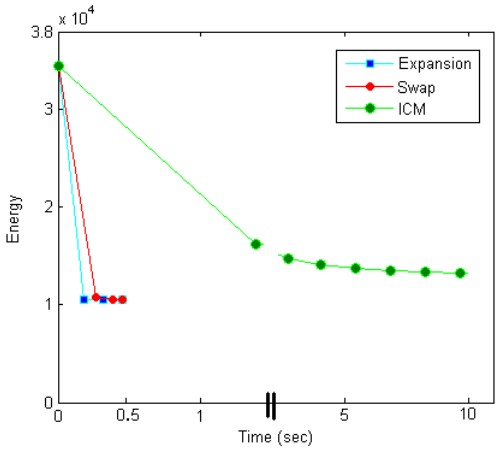


Figure 5. *Comparison of solution energy with respect to runtime. For the experiment, we used a CRF defined over a rectangular grid of* 6000 *random variables with a label set of size* 7*. The pairwise terms of the random field enforced* 8 *connectivity. The energy function used in the experiment had higher order potentials defined over a random number of cliques. These cliques were generated so that each variable of the random field is included in exactly one higher order clique. The graph shows how the energy of the solution obtained from different minimization algorithms changes with time when we use a Robust P^n model with the truncation parameter Q equal to one tenth of the clique size i.e. $Q = 0.1|c|$. It can be seen that expansion and swap algorithms are much faster and produce better solutions than ICM.*

rithms which are able to handle higher order energy functions. Our work attempts to make progress in this direction.

In this paper we introduced a novel family of higher order potentials (Robust P^n model). We showed that energy functions composed of such potentials can be minimized using the powerful graph cut based expansion and swap move algorithms. Our methods for computing the optimal expansion and swap moves are extremely efficient and can handle potentials defined over cliques of thousands of variables. For functions of binary variables, our method is able to produce the globally optimal solution in polynomial time. Our companion paper shows that incorporation of Robust P^n model potentials in the conditional random field model for object segmentation improves results. We believe that these potentials can produce similar improvements for many other labelling problems in Computer Vision.

Up until now, the work on solving higher order potentials using move making algorithms has targeted particular classes of potential functions. Developing efficient large move making for exact and approximate minimization of general higher order energy functions is an interesting and challenging problem for future research.

References

[1] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002. 3, 4

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 1, 2, 3

[3] B. Flach. Strukturelle bilderkennung. Technical report, Universit at Dresden, 2002. 3

[4] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR*, 2005. 3, 4

[5] S. Fujishige. *Submodular functios and optimization*. Annals of Discrete Mathematics, Amsterdam, 1991. 3

[6] H. Ishikawa. Exact optimization for markov random fields with convex priors. *PAMI*, 25:1333–1336, October 2003. 3

[7] P. Kohli, M. Kumar, and P. Torr. P^3 and beyond: Solving energies with higher order cliques. In *CVPR*, 2007. 1, 2, 3

[8] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008. 1, 5, 7

[9] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts?. *PAMI*, 26(2):147–159, 2004. 2, 3, 4, 6

[10] N. Komodakis and G. Tziritas. A new framework for approximate labeling via graph cuts. In *ICCV*, 2005. 2

[11] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic mrfs. In *CVPR*, 2007. 2

[12] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV (2)*, pages 269–282, 2006. 1, 7

[13] V. Lempitsky, C. Rother, and A. Blake. Logcut - efficient graph cut optimization for markov random fields. In *ICCV*, 2007. 3

[14] L. Lovasz. Submodular functions and convexity. In *Mathematical Programming: The State of the Art*, pages 235–257, 1983. 3

[15] J. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. In *IPCO*, pages 240–251, 2007. 3

[16] R. Paget and I. Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, 1998. 1

[17] B. Potetz. Efficient belief propagation for vision using linear constraint nodes. In *CVPR*, 2007. 1

[18] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005. 1, 7

[19] S. Roth and M. Black. On the spatial statistics of optical flow. In *ICCV*, pages 42–49, 2005. 1

[20] D. Schlesinger and B. Flach. Transforming an arbitrary minsum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology, April 2006. 3

[21] J. Shotton, J. Winn, C. Rother, and A. Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV (1)*, pages 1–15, 2006. 7

[22] O. Veksler. Graph cut based optimization for mrfs with truncated convex priors. In *CVPR*, 2007. 3

[23] M. Wainwright, T. Jaakkola, and A. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005. 1, 2

[24] Y. Weiss and W. Freeman. What makes a good model of natural images? In *CVPR*, 2007. 7

[25] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR (1)*, pages 37–44, 2006. 3

[26] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000. 2

Appendix

Theorem 1 The higher order move energy (23) can be transformed into a pairwise energy function by introducing binary meta-variables m_0 and m_1 as:

$$\begin{aligned}\psi_c^m(\mathbf{t}_c) &= \min_{m_0,m_1} r_0(1-m_0)+\theta_\beta m_0\sum_{i\in c_a} w_i(1-t_i)\\ &+ r_1 m_1+\theta_\alpha(1-m_1)\sum_{i\in c_a} w_i t_i-\delta, \end{aligned}\tag{30}$$

where $r_0=\lambda_\alpha+\delta$, $r_1=\lambda_\beta+\delta$, and $\delta=\lambda_{\max}-\lambda_\alpha-\lambda_\beta$.

Proof. The proof is similar to the one for the Robust P^n model. We decompose the function (30) as:

$$\psi_c^m(\mathbf{t}_c)=\mathcal{F}^0(\mathbf{t}_{c_a})+\mathcal{F}^1(\mathbf{t}_{c_a})-\delta \text{ where}\tag{31}$$

$$\begin{aligned}\mathcal{F}^0(\mathbf{t}_{c_a}) &= \min_{m_0} r_0(1-m_0)+f_0^m(\mathbf{t}_{c_a})\theta_\beta m_0 &(32)\\ &= \min_{m_0}(\lambda_\alpha+\delta)(1-m_0)+\theta_\beta m_0 f_0^m(\mathbf{t}_{c_a})\\ &= \min_{m_0}(\lambda_{\max}-\lambda_\beta)(1-m_0)\\ &+ \frac{\gamma_{\max}-\gamma_\beta}{Q_\beta} m_0 f_0^m(\mathbf{t}_{c_a}) &(33)\end{aligned}$$

On substituting the value of $\lambda_{\max}$ and λ_β we get

$$\begin{aligned}\mathcal{F}^0(\mathbf{t}_{c_a}) &= \min_{m_0}(\gamma_{\max}-\gamma_\beta-R_{\alpha\beta}\theta_\beta)(1-m_0)\\ &+ \frac{\gamma_{\max}-\gamma_\beta}{Q_\beta} m_0 f_0^m(\mathbf{t}_{c_a}) &(34)\\ &= \min_{m_0}(\gamma_{\max}-\gamma_\beta)(1-m_0)\\ &+ \frac{\gamma_{\max}-\gamma_\beta}{Q_\beta} m_0(f_0^m(\mathbf{t}_{c_a})+R_{\alpha\beta})-R_{\alpha\beta}\theta_\beta\end{aligned}$$

$$\mathcal{F}^0(\mathbf{t}_{c_a})=\begin{cases}\lambda_{\max}-\lambda_\beta & \text{if } f_0^m(\mathbf{t}_{c_a})>Q_\beta-R_{\alpha\beta}\\ f_0^m(\mathbf{t}_{c_a})\theta_\beta & \text{if } f_0^m(\mathbf{t}_{c_a})\le Q_\beta-R_{\alpha\beta}\end{cases}\tag{35}$$

Similarly, $\mathcal{F}^1(\mathbf{t}_{c_a})=$

$$\begin{cases}\lambda_{\max}-\lambda_\alpha & \text{if } f_1^m(\mathbf{t}_{c_a})>Q_\alpha-R_{\alpha\beta}\\ (W(c_a)-f_0^m(\mathbf{t}_{c_a}))\theta_\alpha & \text{if } f_1^m(\mathbf{t}_{c_a})\le Q_\alpha-R_{\alpha\beta}\end{cases}\tag{36}$$

or $\mathcal{F}^1(\mathbf{t}_{c_a})=$

$$\begin{cases}\lambda_{\max}-\lambda_\alpha & \text{if } f_0^m(\mathbf{t}_{c_a})\le W(c_a)-(Q_\alpha-R_{\alpha\beta})\\ (W(c_a)-f_0^m(\mathbf{t}_{c_a}))\theta_\alpha & \text{if } f_0^m(\mathbf{t}_{c_a})>W(c_a)-(Q_\alpha-R_{\alpha\beta}).\end{cases}\tag{37}$$

Adding equations (35) and (37) and from the constraints $Q_i+Q_j<P, \forall i\neq j\in\mathcal{L}$, we get $\mathcal{F}^0(\mathbf{t}_{c_a})+\mathcal{F}^1(\mathbf{t}_{c_a})=$

$$\begin{cases}\lambda_{\max}-\lambda_\beta+(W(c_a)-f_0^m(\mathbf{t}_{c_a}))\theta_\alpha & \text{if } f_0^m(\mathbf{t}_{c_a})>W(c_a)-\hat{Q}_\alpha\\ f_0^m(\mathbf{t}_{c_a})\theta_\beta+\lambda_{\max}-\lambda_\alpha & \text{if } f_0^m(\mathbf{t}_{c_a})\le\hat{Q}_\beta,\\ \lambda_{\max}-\lambda_\alpha+\lambda_{\max}-\lambda_\beta & \text{otherwise.}\end{cases}\tag{38}$$

where $\hat{Q}_k=Q_k-R_{\alpha\beta}$, $k\in\{\alpha,\beta\}$. Substituting this in (31) and simplifying we get:

$$\psi_c^m(\mathbf{t}_{c_a})=\begin{cases}K_\alpha & \text{if} & f_0^m(\mathbf{t}_{c_a})>W(c_a)-\hat{Q}_\alpha\\ K_\beta & \text{if} & f_0^m(\mathbf{t}_{c_a})\le\hat{Q}_\beta\\ \lambda_{\max} & \text{otherwise}\end{cases}\tag{39}$$

where $K_\alpha=\lambda_\alpha+(W(c_a)-f_0^m(\mathbf{t}_{c_a}))\theta_\alpha$, and $K_\beta=\lambda_\beta+f_0^m(\mathbf{t}_{c_a})\theta_\beta$. This is the same as (23). □

Theorem 2 The expansion move energy (26) can be transformed into the pairwise function:

$$\begin{aligned}\psi_c^m(\mathbf{t}_c) &= \min_{m_0,m_1} r_0(1-m_0)+\theta_d m_0\sum_{i\in c_d} w_i(1-t_i)\\ &+ r_1 m_1+\theta_\alpha(1-m_1)\sum_{i\in c} w_i t_i-\delta.\end{aligned}\tag{40}$$

where $r_0=\lambda_\alpha+\delta$, $r_1=\lambda_d+\delta$, and $\delta=\lambda_{\max}-\lambda_\alpha-\lambda_d$.

Proof. We decompose the move energy (40) as:

$$\psi_c^m(\mathbf{t}_c)=\mathcal{F}^0(\mathbf{t}_{c_d})+\mathcal{F}^1(\mathbf{t}_c)-\delta \text{ where}\tag{41}$$

$$\begin{aligned}\mathcal{F}^0(\mathbf{t}_{c_d}) &= \min_{m_0} r_0(1-m_0)+f_0^m(\mathbf{t}_{c_d})\theta_d m_0 &(42)\\ &= \min_{m_0}(\lambda_\alpha+\delta)(1-m_0)+\theta_d m_0 f_0^m(\mathbf{t}_{c_d}) &(43)\\ &= \min_{m_0}(\gamma_{\max}-\gamma_d-R_d\theta_d)(1-m_0)\\ &+ \frac{\gamma_{\max}-\gamma_d}{Q_d} m_0 f_0^m(\mathbf{t}_{c_d}) \quad (\text{Recall } R_d=W(c-c_d))\\ &= \min_{m_0}(\gamma_{\max}-\gamma_d)(1-m_0)\\ &+ \frac{\gamma_{\max}-\gamma_d}{Q_d} m_0(f_0^m(\mathbf{t}_{c_d})+R_d)-R_d\theta_d\\ &= \begin{cases}\lambda_{\max}-\lambda_d & \text{if } f_0^m(\mathbf{t}_{c_d})>Q_d-R_d\\ f_0^m(\mathbf{t}_{c_d})\theta_d & \text{if } f_0^m(\mathbf{t}_{c_d})\le Q_d-R_d\end{cases}. &(44)\end{aligned}$$

Similarly,

$$\begin{aligned}\mathcal{F}^1(\mathbf{t}_c) &= \min_{m_1} r_1 m_1+f_1^m(\mathbf{t}_c)\theta_\alpha(1-m_1) &(45)\\ &= \min_{m_1}(\lambda_d+\delta)m_1+\theta_\alpha(1-m_1)f_1^m(\mathbf{t}_c) &(46)\\ &= \min_{m_1}(\gamma_{\max}-\gamma_\alpha)m_1\\ &+ \frac{\gamma_{\max}-\gamma_\alpha}{Q_\alpha}(1-m_1)f_1^m(\mathbf{t}_c) &(47)\\ &= \begin{cases}\lambda_{\max}-\lambda_\alpha & \text{if } f_1^m(\mathbf{t}_c)\ge Q_\alpha\\ f_1^m(\mathbf{t}_c)\theta_\alpha & \text{if } f_1^m(\mathbf{t}_c)<Q_\alpha.\end{cases} &(48)\\ &= \begin{cases}\lambda_{\max}-\lambda_\alpha & \text{if } f_0^m(\mathbf{t}_c)\le P-Q_\alpha\\ f_1^m(\mathbf{t}_c)\theta_\alpha & \text{if } f_0^m(\mathbf{t}_c)>P-Q_\alpha\end{cases} &(49)\end{aligned}$$

Adding (44) and (49) and using the relations[4]

$$f_0^m(t_{c_d})\le Q_d-R_d \implies f_0^m(\mathbf{t}_c)\le P-Q_d\tag{50}$$

$$f_0^m(\mathbf{t}_c)>P-Q_d \implies f_0^m(t_{c_d})>Q_d-R_d\tag{51}$$

we get: $\mathcal{F}^0(\mathbf{t}_{c_a})+\mathcal{F}^1(\mathbf{t}_{c_a})=$

$$\begin{cases}\lambda_{\max}-\lambda_d+(P-f_0^m(\mathbf{t}_c))\theta_\alpha & \text{if } f_0^m(\mathbf{t}_c)>P-Q_\alpha\\ f_0^m(\mathbf{t}_{c_d})\theta_d+\lambda_{\max}-\lambda_\alpha & \text{if } f_0^m(\mathbf{t}_{c_d})\le Q_d-R_d\\ \lambda_{\max}-\lambda_\alpha+\lambda_{\max}-\lambda_d & \text{otherwise.}\end{cases}\tag{52}$$

Substituting in (41) and simplifying we get $\psi_c^m(\mathbf{t}_c,\mathbf{t}_{c_d})=:$

$$\begin{cases}\lambda_\alpha+(P-f_0^m(\mathbf{t}_c))\theta_\alpha & \text{if } f_0^m(\mathbf{t}_c)>P-Q_\alpha\\ \lambda_d+f_0^m(\mathbf{t}_{c_d})\theta_d & \text{if } f_0^m(\mathbf{t}_{c_d})\le Q_d-R_d\\ \lambda_{\max} & \text{otherwise}\end{cases}\tag{53}$$

which is the same as (26). □

[4] These relations are derived from the constraints $Q_i+Q_j<P, \forall i\neq j\in\mathcal{L}$ and $W(s)\le P, \forall s\subseteq c$.