Information Sciences 328 (2016) 158–171

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Discrete particle swarm optimization for high-order graph matching

Maoguo Gong [a,*], Yue Wu [a], Qing Cai [a], Wenping Ma [a], A.K. Qin [b], Zhenkun Wang [a], Licheng Jiao [a]

[a] *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China*
[b] *School of Computer Science and Information Technology, RMIT University, Melbourne, Vic. 3001, Australia*

ARTICLE INFO

Article history:
Received 29 October 2014
Revised 10 August 2015
Accepted 23 August 2015
Available online 28 August 2015

Keywords:
High-order graph matching
Particle swarm optimization
Evolutionary algorithm

ABSTRACT

High-order graph matching aims at establishing correspondences between two sets of feature points using high-order constraints. It is usually formulated as an NP-hard problem of maximizing an objective function. This paper introduces a discrete particle swarm optimization algorithm for resolving high-order graph matching problems, which incorporates several re-defined operations, a problem-specific initialization method based on heuristic information, and a problem-specific local search procedure. The proposed algorithm is evaluated on both synthetic and real-world datasets. Its outstanding performance is validated in comparison with three state-of-the-art approaches.

1. Introduction

Many challenging tasks in computer vision can be formulated as graph matching problems, such as shape matching [44], object detection [38], image classification [22], and object tracking [4]. These tasks typically involve the extraction of feature points from images and the establishment of the correspondences between the sets of feature points extracted from different images. Many research efforts have been devoted to addressing the latter sub-task using graph matching techniques. Our work focuses on finding the reliable correspondences between two points through their relation information with other points.

Graph matching can be formulated as an optimization problem with respect to an objective function that evaluates the machining accuracy. This function typically measures the similarity between two graphs in terms of certain graph features, e.g., features with respect to nodes, features with respect to edges linking two nodes, etc. First-order graph matching methods only use node features to establish the correspondence [21]. Such methods may fail when many nodes from the same graph have similar features due to the ignorance of the spatial relationship information between nodes. Second-order graph matching methods, as the one proposed in [24], consider the features with respect to the edges linking two graph nodes which can capture some structural information of the graph. Second-order information from graph edges can improve the accuracy achieved by using first-order information from graph nodes [10,17]. However, second-order graph matching is not robust to some kinds of deformations under which the matching the accuracy may drop dramatically. To address such drawbacks, high-order graph matching methods had been proposed [39], which incorporate high-order constraints by considering the relationships among a tuple of graph nodes. In most cases, third-order constraints with respect to triplets of graph nodes are applied. High-order graph matching

* Corresponding author. Tel.: +86 029 88202661; fax: +86 029 88201023.
E-mail address: gong@ieee.org (M. Gong).

http://dx.doi.org/10.1016/j.ins.2015.08.038

methods use hyper-edges to represent high-order constraints. The features with respect to hyper-edges are usually more robust to deformations. Zass and Shashua presented a novel probabilistic model for high-order graph matching [39]. Duchenne *et al.* proposed a tensor-based method [13]. Lee *et al.* developed an improved reweighed random walks algorithm [23]. Due to the large demand for the affinity tensor, Park*et al.* presented a fast and efficient computational approach [30]. High-order graph matching typically corresponds to a highly complex optimization problem which challenges most existing approaches as described above. Therefore, more effective optimization algorithms are in high demand.

Particle swarm optimization (PSO) is one of the most popular swarm intelligence technique which was proposed by Kennedy and Eberhart in 1995 [14,18]. It is inspired by the social behavior, such as bird flocking or fish schooling. A swarm is typically composed of a fixed number of particles. Each particle is associated with a position term and a velocity term. The new position of a particle is determined by its current position and velocity. Each particle keeps learning from the previous search experiences from both itself and its neighboring particles to search for the optimal solution. PSO is simple but powerful for solving many complex optimization problems [14,20,34]. Many PSO variants have been proposed to improve the performance of the basic version. [2,7,40,41]. CLPSO [26] uses a novel learning strategy to enhance the diversity so as to prevent discourage premature convergence when solving multimodal problems. ALC–PSO [6] assigns the leader of the swarm with a growing age and a lifespan to escape from local optima and thus prevent premature convergence. SRPSO [36] uses the self-regulating inertia weight and the self-perception on the global search direction to achieve faster convergence and better solutions.

PSO has achieved great success in continuous optimization problems [11,25,28,43], Kennedy and Eberhart also proposed a discrete binary version of PSO to solve discrete problems [19]. By re-defining the position and velocity of the particle, continuous PSO is turned into discrete PSO to solve discrete optimization problems. Chen *et al.* proposed a set-based PSO (S–PSO) for discrete optimization [5] in which the position and velocity of the particle were re-defined using the concepts of set and probability. Discrete PSO (DPSO) has been successfully applied to many discrete optimization tasks, such as traveling salesman problems (TSPs) [35,37], multidimensional knapsack problems (MKPs) [1,16], job shop scheduling problems (JSPs) [32,42], complex network clustering problems [3,15], and image matching problems [8,31]. These problems all have their respective challenges and are hard to optimize, but they can be effectively solved by DPSO.

In this paper, we proposes a DPSO algorithm to address challenging high-order graph matching problems. There are several reasons that motivate us to choose PSO. First, PSO has the fast convergence speed toward optima. Second, discrete PSO has demonstrated strong efficacy in solving challenging discrete optimization problems. Third, PSO has just a few parameters, making it easy to implement.

High-order graph matching is typically formulated as a NP-hard problem of maximizing an objective function. The optimal solution to this NP-hard problem is practically unachievable so that most existing methods only provided approximate solutions [23]. Meta-heuristic algorithms are an effective technique for addressing NP-hard problems [12]. Our proposed DPSO is a meta-heuristic algorithm, which incorporates several re-defined PSO operations, a problem-specific initialization method based on heuristic information, and a problem-specific local search procedure. Experimental results show that our proposed algorithm is promising for high-order graph matching.

The rest of this paper is organized as follows. Section 2 introduces the related background. In Section 3, the proposed method is described in detail. The performance of the proposed method is evaluated and compared with those of three state-of-the-art approaches on both synthetic and real-world datasets. Section 5 draws the conclusions.

2. Related background

2.1. High-order graph matching

High-order graph matching is an extension of spectral matching, which imposes the constraints on the relationships among a tuple of nodes [24]. Given two graphs G^P and G^Q, $G^P = (V^P, E^P, A^P)$ and $G^Q = (V^Q, E^Q, A^Q)$. V is the set of nodes v_i, E is the set of hyper-edges e_i with each formed by a tuples of nodes, and A is the set of attributes a_i characterizing hyper-edge e_i. Our goal is to establish the correspondences between $V^P$and V^Q constrained on the consistency between A^P and A^Q. Accordingly, the problem is to find an assignment matrix Z of size $N_P \times N_Q$, N_P is the number of nodes in G^P and N_Q is the number of nodes in G^Q. Z is a binary matrix, where Z_{i_1,i_2} is equal to 1 when $v^P_{i_1}$ is matched to $v^Q_{i_2}$ and 0 otherwise. Each node in graph G^P can only be matched to one point in graph G^Q and vice versa. We call this a one-to-one constraint, which is common in graph matching.

Given third-order constraints, each hyper-edge $e^P_{i_1,j_1,k_1}$ is formed by a triplet of nodes $v^P_{i_1}$, $v^P_{j_1}$, $v^P_{k_1}$, and characterized by attribute $a^P_{i_1,j_1,k_1}$. $H_{i_1,i_2,j_1,j_2,k_1,k_2}$ represents the affinity between $e^P_{i_1,j_1,k_1}$ and $e^Q_{i_2,j_2,k_2}$ which constitutes the affinity tensor H [13]. Tensor H is nonnegative and symmetric, such that

$$H_{i_1,i_2,j_1,j_2,k_1,k_2} \geq 0$$

$$\begin{aligned} H_{i_1,i_2,j_1,j_2,k_1,k_2} &= H_{i_1,i_2,k_1,k_2,j_1,j_2} = H_{j_1,j_2,i_1,i_2,k_1,k_2} \\ &= H_{k_1,k_2,i_1,i_2,j_1,j_2} = H_{k_1,k_2,j_1,j_2,i_1,i_2} \\ &= H_{j_1,j_2,k_1,k_2,i_1,i_2} \end{aligned}$$