Optim Lett (2012) 6:315–329
DOI 10.1007/s11590-010-0259-x

ORIGINAL PAPER

Globally solving a nonlinear UAV task assignment problem by stochastic and deterministic optimization approaches

Hoai An Le Thi · Duc Manh Nguyen · Tao Pham Dinh

Received: 11 October 2010 / Accepted: 25 October 2010 / Published online: 25 November 2010

Abstract In this paper, we consider a task allocation model that consists of assigning a set of *m* unmanned aerial vehicles (UAVs) to a set of *n* tasks in an optimal way. The optimality is quantified by target scores. The mission is to maximize the target score while satisfying capacity constraints of both the UAVs and the tasks. This problem is known to be NP-hard. Existing algorithms are not suitable for the large scale setting. Scalability and robustness are recognized as two main issues. We deal with these issues by two optimization approaches. The first approach is the Cross-Entropy (CE) method, a generic and practical tool of stochastic optimization for solving NP-hard problem. The second one is Branch and Bound algorithm, an efficient classical tool of global deterministic optimization. The numerical results show the efficiency of our approaches, in particular the CE method for very large scale setting.

Keywords UAV · Task assignment problem · Stochastic programming · Binary nonlinear programming · Cross-entropy (CE) method · Brand and bound algorithm

H. A. Le Thi (✉)
Laboratory of Theoretical and Applied Computer Science, UFR MIM,
Paul Verlaine University, Metz, Ile du Saulcy, 57045 Metz, France
e-mail: lethi@univ-metz.fr

D. M. Nguyen · T. Pham Dinh
Laboratory of Modelling, Optimization and Operations Research,
National Institute for Applied Sciences, Rouen, 76801 Saint-Etienne-du-Rouvray, France
e-mail: duc.nguyen@insa-rouen.fr

T. Pham Dinh
e-mail: pham@insa-rouen.fr

1 Introduction

The use of unmanned aerial vehicles (UAVs) for various military missions has received growing attention in the past years. Apart from the obvious advantage of not placing human life at risk, the lack of a human pilot enables considerable weight savings and lower costs. On the other hand, UAVs provide an opportunity for new operational paradigms. However, to realize these advantages, UAVs must have a high level of autonomy and capacity to work cooperatively in groups. In this context, several algorithms dealing with the problem of commanding multiple UAVs to cooperatively perform multiple tasks have been developed. The aim is to assign specific tasks and flyable trajectories to each vehicle to maximize the group performance. The intrinsic uncertainty imbedded in military operations makes the problem more challenging. Scalability and robustness are recognized as two main issues. Also, to allow implementation, the developed algorithms must be solved in real time.

Extensive research has been done recently in this field [4,7,9,12–18,20,21,27–30]. In [4,17,20], task allocation has been formulated in the form of Mixed-Integer Linear Programming (MILP). In this approach, the problem is solved as a deterministic optimization problem with known parameters. Since the MILP is NP-hard, it suffers from poor scalability although the solutions preserve global optimality [19]. Moreover, military situations are in general dynamic and uncertain because of the UAV's sensing limitation and adversarial strategies. Thus, replanning is necessary whenever the information is updated. Heuristics and ad-hoc methods have been considered during replanning in [14,17]. On the other hand, uncertainty is considered via optimization parameters, and risk management techniques in finance are utilized (see e.g. [21,28]). In [21], a nonlinear integer programming problem is formulated where a risk measure by conditional value-at-risk is considered as constraint. In [28], a robust approach using the Soyster formulation on the expectation of the target scores is investigated. These approaches are based on solving hard combinatorial optimization problems and then scalability is still a big issue. An alternative approach dealing with uncertainties consists of formulating a stochastic optimal control problem by using the method of Model Predictive Control (MPC) [6,27].

In this paper, we are interested in task allocation models where we seek to assign a set of m UAVs to a set of n tasks in an optimal way. The optimality is quantified by target scores. The mission is to maximize the target score while satisfying capacity constraints of both the UAVs and the tasks.

The scoring scheme defining effectiveness in our work is a nonlinear function. More precisely, our considered problem is a mixed integer nonlinear programming problem for which the classical MILP solution method can not be used. We propose two approaches to tackle it. The first approach is the Cross-Entropy (CE) method, a simple generic and practical tool of stochastic optimization for solving NP-hard problem. The second one is the Branch and Bound algorithm, an efficient classical tool of global deterministic optimization.

The CE method was originally developed in [22] for an adaptive networks, where an adaptive variance minimization algorithm for estimating probabilities of rare events for stochastic networks was presented. It was modified in [23,24] to solve optimization problems. Several recent publications demonstrate the power of the CE method as simple and efficient approach for many applications such as telecommunication systems, buffer allocation, vehicle routing, DNA sequence alignment, Machine Learning, etc. It has been proved that this method is particularly relevant for solving "hard" combinatorial optimization problems. In fact, when deterministic methods failed to find the optimal solution within a reasonable time, in most cases the CE method allows to find a fairly good solution more quickly. This motivates us to investigate the CE method for large scale UAV Task Assignment Problem. For measuring the efficiency of the CE method and globally solving the considered problem, we develop a Brand and Bound (B&B) algorithm and compare the two methods.

The rest of paper is organized as follows. In Sect. 2, we describe the problem and give its mathematical formulation. Section 3 is dedicated to the description of CE method and its application for solving the considered problem. The B&B is presented in Sect. 4 while the numerical experiments are reported in Sect. 5. Finally we conclude the paper by Sect. 6.

2 Problem statement

Let V and T be the sets of m UAVs and n targets, respectively. The scoring scheme defining effectiveness is based on the definition of target score. Each target j has an associated score based on the task success probability r_j and a weight w_j measuring the importance of the target. The probability that the task will be successfully carried out for that target depends on y_j, the number of UAVs which have been assigned to the target j, in the following way:

$$1 - (1 - r_j)^{y_j}.$$

A target score is computed as the product of the success probability and its weight:

$$g_j(y_j) = w_j(1 - (1 - r_j)^{y_j}), \tag{1}$$

and the UAVs group effectiveness is simply the sum of all individual target scores: $\sum_{j \in T} g_j(y_j)$. Then the goal is to maximize the UAVs group effectiveness.

Let z_{ij}, for $i \in V = \{1, \ldots, m\}$ and $j \in T = \{1, \ldots, n\}$, be the decision variable defined by: z_{ij} is equal to 1 when the UAV i is assigned to the target j, and 0 otherwise. An entry of $m \times n$ adjacency matrix A, a_{ij}, indicates which target each UAV can be assigned. So, the number y_j can be computed as $y_j = \sum_{i \in V} a_{ij} z_{ij},\ \ j \in T$.

The mathematical model of this problem can be written as follows:

$$\begin{cases} \max\limits_{x,y} \sum\limits_{j \in T} w_j(1 - (1 - r_j)^{y_j}) \\ s.t. \quad y_j = \sum\limits_{i \in V} a_{ij} z_{ij}, \quad j \in T, \\ \sum\limits_{j \in T} z_{ij} = 1, \quad i \in V, \\ z_{ij} \in \{0, 1\}, \quad i \in V,\ j \in T. \end{cases} \tag{2}$$

The second constraints ensure that each UAV i is used for only one task. This problem is an integer nonlinear programming which is known to be very hard.

3 A Cross-Entropy algorithm for solving the UAV task assignment problem (2)

3.1 An introduction to Cross-Entropy method

The CE method is a relatively new method for solving both continuous multi-extremal and combinatorial optimization problems. It was originally developed in the rare-event estimation framework [22] as an adaptive importance sampling scheme for estimating rare event probabilities via simulation. This approach was afterward modified in [23,24] for solving both continuous multi-extremal and combinatorial optimization problems. The main idea of the CE method is the construction of a random sequence of solutions which converges probabilistically to the optimal or near-optimal solution. It involves the following two iterative phases:

1. Generation of a sample of random data (trajectories, vectors, etc.) according to a specified random mechanism.
2. Updating the parameters of the random mechanism, typically parameters of pdfs (probability density functions), on the basis of the data, to produce a "better" sample in the next iteration.

Unlike most of the stochastic algorithms for optimization which are based on local search, the CE method is a global random search procedure. The CE method was successfully applied to various problems such as the traveling salesman problem [23], the bipartition problem [23], the maximal cut problem [25], the image matching [10], the image segmentation [11], etc.

For a comprehensive overview and history of the CE method, the reader is referred to [26]. For the sake of completion we present below the generic CE scheme for combinatorial optimization problems.

Consider the problem of minimizing the function S over a finite set $\mathscr{X}$, say

$$\gamma^* = \min_{x \in \mathscr{X}} S(x). \tag{3}$$

The starting point in the methodology of the CE method applied to (3) is to associate an estimation problem with the optimization problem (3). To this end one defines a collection of indicator functions $I_{\{S(x) \le \gamma\}}$ on $\mathscr{X}$ for various thresholds or levels $\gamma \in \mathbb{R}$. Next, let $\{f(\cdot; v), v \in V\}$ be a family of (discrete) probability density functions (pdfs) on $\mathscr{X}$, parameterized by a real-valued (vector) v.

For some $u \in V$, we consider the *Associated Stochastic Problem* (ASP):

$$\ell(\gamma) = \mathbb{P}_u(S(x) \le \gamma) = \sum_{x \in \mathscr{X}} I_{\{S(x) \le \gamma\}} f(x; u) = \mathbb{E}_u I_{\{S(x) \le \gamma\}}, \tag{4}$$

where $\mathbb{P}_u$ is the probability measure under which the random state $\mathscr{X}$ has the pdf $f(\cdot; u)$, and $\mathbb{E}_u$ denotes the corresponding expectation operator. The idea of CE method is to construct simultaneously two sequences of levels $\widehat{\gamma}_1, \widehat{\gamma}_2, \ldots, \widehat{\gamma}_T$ and parameters (vectors) $\widehat{v}_1, \widehat{v}_2, \ldots, \widehat{v}_T$ such that $\widehat{\gamma}_T$ is close to the optimal γ^*, and $\widehat{v}_T$ is such that the corresponding density assigns high probability mass to the collection of states that give a low value. More specifically, one initializes by setting $v_0 = u$ and choosing a not very small quantity θ, and then proceeds as follows:

1. **Adaptive updating of γ_t.** For a fixed v_t, let γ_t be the θ-quantile of $S(X)$ under v_{t-1}. That is, γ_t satisfies

$$\mathbb{P}_{v_{t-1}}(S(X) \geq \gamma_t) \geq 1 - \theta, \quad (5)$$

$$\mathbb{P}_{v_{t-1}}(S(X) \leq \gamma_t) \geq \theta, \quad (6)$$

where $X \sim f(\cdot; v_{t-1})$.
A simple estimator of γ_t, denotes $\widehat{\gamma}_t$, can be obtained by drawing a random sample $X^1, X^2, \ldots, X^N$ from $f(\cdot; v_{t-1})$. Suppose that $S(X^{\sigma(1)}) \leq S(X^{\sigma(2)}) \leq \cdots \leq S(X^{\sigma(N)})$, where σ is a permutation of the set $\{1, \ldots, N\}$. Estimate the θ-quantile of $S(X)$ as

$$\widehat{\gamma}_t = S_{\lfloor \theta N \rfloor}. \quad (7)$$

2. **Adaptive updating of v_t.** For a fixed γ_t and v_{t-1}, derive v_t by minimizing the Kullback-Leibler distance, or equivalent to solving the program

$$\max_v \mathbb{E}_{v_{t-1}} I_{\{S(X) \leq \gamma_t\}} \ln f(X; v). \quad (8)$$

The stochastic counterpart of (8) is as follows: for fixed $\widehat{\gamma}_t$ and $\widehat{v}_{t-1}$ the estimate of v_{t-1}, derive $\widehat{v}_t$ from the solution of following program

$$\max_v D(v) := \frac{1}{N} \sum_{i=1}^{N} I_{\{S(X^i) \leq \gamma_t\}} \ln f(X^i; v). \quad (9)$$

In typical applications, the function D is concave and differentiable with respect to v, and thus updating Eq. (9) is equivalent to solving the following system of equations:

$$\frac{1}{N} \sum_{i=1}^{N} I_{\{S(X^i) \leq \gamma_t\}} \nabla \ln f(X^i; v) = 0, \quad (10)$$

where the gradient is with respect to v.

CE Algorithm for combinatorial optimization

1. Choose $\widehat{v}_0$, and $0 < \theta < 1$. Set $t = 1$.
2. Generate N samples $X^1, X^2, \ldots, X^N$ according to $f(\cdot; \widehat{v}_{t-1})$, and compute θ-quantile $\widehat{\gamma}_t$ of S according to (7).

3. Use the same samples $X^1, X^2, \ldots, X^N$ to solve the stochastic programming problem (9). Denote the solution by $\widehat{v}_t$.
4. If for some $t \geq d$, say $d = 5$ such that

$$\widehat{\gamma}_t = \widehat{\gamma}_{t-1} = \cdots = \widehat{\gamma}_{t-d},$$

then **stop**; otherwise set $t = t + 1$, reiterate from step 2.

For the convergence analysis of the CE method we refer to [5,8,26].

3.2 A Cross-Entropy method for solving problem (2)

We first rewrite the problem (2) in the form:

$$\begin{cases} \min\limits_{x} f(z) := \sum\limits_{j \in T} w_j \left(1 - r_j\right)^{\sum_{i \in V} a_{ij} z_{ij}} \\ s.t. \quad \sum\limits_{j \in T} z_{ij} = 1, \quad \forall i \in V, \\ z_{ij} \in \{0, 1\}, \quad \forall i \in V, j \in T. \end{cases} \tag{11}$$

Denote by Z the feasible set of (11), say

$$Z := \left\{ z = (z_{ij}) \in \{0, 1\}^{mn} : \sum_{j \in T} z_{ij} = 1, i \in V \right\}.$$

It is clear that each variable $z \in Z$ is identical to an assignment mapping $m_z : \{1, \ldots, m\} \longmapsto \{1, \ldots, n\}$, i.e., $m_z(i) = j$ iff $z_{ij} = 1$. Then, the set Z is identical to the set:

$$\mathscr{X} = \{x = (x_1, \ldots, x_m) : x_i \in \{1, \ldots, n\} \text{ is the target assigned to UAV } i\}. \tag{12}$$

The CE algorithm draws particular assignment of UAVs to targets that will be evaluated and then selected, in order to obtain a drawing law which will converge toward the optimal assignment. First, we must choose a family of pdf, $f(\cdot; v)$, describing a probability choice of x.

A discrete probability law $p(j|i)$ is associated to each UAV i. It represents the probability to assign the UAV i to the target j. These probabilities are summarized by the matrix M:

$$M = \begin{pmatrix} p(1|1) & p(2|1) & \cdots & p(n|1) \\ p(1|2) & p(2|2) & \cdots & p(n|2) \\ & & \cdots & \\ p(1|m) & p(2|m) & \cdots & p(n|m) \end{pmatrix}.$$

Note that, we must have

$$\sum_{j=1}^{n} p^M(j|i) = 1.$$

Let $X = (x_1, x_2, \ldots, x_m)$ be the random assignment vector of UAVs to targets, x_i is the target assigned to UAV i for the draw. The probability of drawing the vector according to M is

$$p(X) = \prod_{i=1}^{m} p(x_i|i),$$

where $p(x_i|i)$ is the coefficient in the column x_i and the row i of matrix M.

In each iteration, suppose that X^k, $X^k = (x_1^k, \ldots, x_i^k, \ldots, x_m^k)$, $k = 1, 2, \ldots, N$ are the samples drawn. The $H = \lfloor \theta N \rfloor$ best samples, according to the objective function f, are selected to update M. Denoting $\{X^1, X^2, \ldots, X^H\}$ as the H "best" vectors among the draws $\{X^1, X^2, \ldots, X^N\}$. Minimizing the Kullback–Leibler distance leads to the following optimization problem:

$$\begin{cases} \max\limits_{p(j|i)} \mathscr{K} := \sum\limits_{h=1}^{H} \ln\left(\prod\limits_{i=1}^{m} p(x_i^h|i)\right) \\ s.t. \quad \sum\limits_{j=1}^{n} p(j|i) = 1, \quad i = 1, \ldots, m, \\ p(j|i) \geq 0, \quad i = 1, \ldots, m, j = 1, \ldots, n. \end{cases} \tag{13}$$

We first rewrite the objective function as follows

$$\begin{aligned} \mathscr{K} &= \sum_{h=1}^{H} \ln\left(\prod_{i=1}^{m} p\left(x_i^h|i\right)\right) = \sum_{i=1}^{m}\sum_{h=1}^{H} \ln\left(p\left(x_i^h|i\right)\right) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} \text{card}\{h \in \{1, \ldots, H\} : x_i^h = j\} \ln(p(j|i)) \end{aligned}$$

Denoting

$$u_{ji} = p(j|i), b_{ji} = \text{card}\{h \in \{1, \ldots, H\} : x_i^h = j\}.$$

The program (13) is equivalent to

$$\begin{cases} \min\limits_{u_{ji}} \left(-\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} b_{ji} \ln(u_{ji})\right) \\ s.t. \quad \sum\limits_{j=1}^{n} u_{ji} = 1, \quad i = 1, \ldots, m, \\ u_{ji} \geq 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n. \end{cases} \tag{14}$$

Since it is a convex programming problem, we consider the Karusk–Kuhn-Tucker (KKT) condition of the program (14):

$$\begin{cases} -\frac{b_{ji}}{u_{ji}} + \lambda_i - \mu_{ji} = 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n, \\ \lambda_i \left(\sum_{j=1}^{n} u_{ji} - 1 \right) = 0, \quad i = 1, \ldots, m, \\ \mu_{ji} u_{ji} = 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n, \\ \lambda_i \geq 0, \quad i = 1, \ldots, m, \\ \mu_{ji} \geq 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n. \end{cases}$$

Then by solving the KKT condition, we get the updating formula of matrix M as follows

$$p(j|i) := \frac{\text{card}\{h \in \{1, \ldots, H\} : x_i^h = j\}}{H}.$$

The CE algorithm for solving the Problem (11):

Step 1. Initialize $M = M_0 = (p_0(j|i))_{m \times n}$ a uniform distribution, i. e.,

$$p_0(j|i) = \frac{1}{n}, \quad i = 1, \ldots, m, \ j = 1, \ldots, n,$$

and choose $\theta \in (0, 1)$,

Step 2. Draw N samples $X^1, X^2, \ldots, X^N$ according to M. Compute $f(X^k)$, $k = 1, 2, \ldots, N$,

Step 3. Sort the sequence $\{f(X^k)\}_{k=1}^N$ in the increasing orders. Let $f(X^{\sigma(1)}) \leq f(X^{\sigma(2)}) \leq \cdots \leq f(X^{\sigma(N)})$, where σ is a permutation of the set $\{1, 2, \ldots, N\}$. Set $H = \lfloor \theta N \rfloor$, then choose H best draws $X^{\sigma(1)}, X^{\sigma(2)}, \ldots, X^{\sigma(H)}$,

Step 4. Update M by the formula

$$p(j|i) := \frac{\text{card}\{h \in \{1, 2, \ldots, H\} : \quad x_i^{\sigma(h)} = j\}}{H},$$

Step 5. Iterate step 2, 3, 4 until convergence.

In practice, we should choose M_0 in the Step 1 as follows:

$$p_0(j|i) = \begin{cases} \frac{1}{n_i} & \text{if } a_{ij} = 1, \\ 0 & \text{if } a_{ij} = 0, \end{cases}$$

where $n_i = \sum_{j=1}^n a_{ij}, i = 1, \ldots, m.$

4 A Branch and Bound algorithm

For globally solving the considered problem, and for measuring the quality of our CE algorithm, we develop a global approach based on classical Branch and Bound (B&B) scheme. The lower bounds are computed by solving the relaxed problem:

$$\min \left\{ f(z) = \sum_{j=1}^{n} w_j (1-r_j)^{\sum_{i=1}^{m} a_{ij} z_{ij}} : z \in K \right\}, \tag{15}$$

where K is nonempty bounded polyhedral convex set in $\mathbb{R}^{mn}$.

Since this problem is convex, it can be solved by standard solvers for convex programming. Here, we use DCA [1–3] to solve it. DCA is an efficient approach for DC (Difference of Convex functions) programming problems. It addresses a general DC program of the form

$$\alpha := \inf \left\{ f(z) := g(z) - h(z) : z \in \mathbb{R}^n \right\} \quad (P_{dc})$$

where g and h are convex, lower semicontinuous proper functions. DC programs with closed convex set constraints C can be cast into (P_{dc}) by adding χ_C, the indicator function on C, with g. ($\chi_C(z) = 0$ if $z \in C$, $+\infty$ otherwise).

The DC (Difference of Convex functions) programming and DCA (DC Algorithms) constitute the backbone of Nonconvex Programming and Global Optimization. They were introduced by Tao in 1985 and extensively developed by An and Tao since 1994 to become now classic and increasingly popular. It is clear that convex programs are *false* DC programs for which DCA can be used. On the other hand, with suitable DC decompositions, DCA applied to convex programs permits to find again standard optimization methods for convex programming.

The idea of DCA is quite simple: each iteration k one linearizes the concave part $-h$ and then solve the resulting convex program. More precisely, DCA consists of computing at each iteration k

$$y^k \in \partial h(z^k), \quad z^{k+1} \in \arg\min_{z \in \mathbb{R}^n} \left\{ g(z) - h(z^k) - \langle z - z^k, y^k \rangle \right\} \; (P_k).$$

For solving (15) we use the following DC decomposition:

$$f(z) = \sum_{j=1}^{n} w_j (1-r_j)^{\sum_{i=1}^{m} a_{ij} z_{ij}} = g(z) - h(z), \tag{16}$$

where

$$g(z) := \frac{\lambda}{2}\|z\|^2; \quad h(z) := \frac{\lambda}{2}\|z\|^2 - \sum_{j=1}^{n} w_j (1-r_j)^{\sum_{i=1}^{m} a_{ij} z_{ij}}. \tag{17}$$

Here, λ takes the associated value such that h is convex.

Description of the DCA for problem (15)

Step 1. Initialization: Choose $z^0 \in K$, $\varepsilon_1 > 0$, and $\varepsilon_2 > 0$. Set $k = 0$.
Step 2. Compute $y^k = \nabla h(z^k)$, with

$$y_{ij}^k = \lambda z_{ij}^k - w_j \log(1-r_j) a_{ij} (1-r_j)^{\sum_{i=1}^{m} a_{ij} z_{ij}^k}$$

for $i = 1, \ldots, m$, $j = 1, \ldots, n$.
Step 3. Compute z^{k+1} by solving the convex quadratic problem

$$\begin{cases} \min_z \left(g(x) - \langle z, y^k \rangle\right) \\ s.t. \quad z \in K. \end{cases}$$

Step 4. Iterate Step 2 and 3 until

$$\left| f\left(z^{k+1}\right) - f(z^k) \right| \le \varepsilon_1 \left(1 + \left| f\left(z^{k+1}\right) \right| \right)$$
$$\text{or} \quad \left\| z^{k+1} - z^k \right\| \le \varepsilon_2 \left(1 + \left\| z^{k+1} \right\| \right).$$

Let Ω be the set defined by

$$\Omega := \left\{ z = (z_{ij}) \in [0,1]^{mn} : \sum_{j \in T} z_{ij} = 1, i \in V \right\}.$$

B&B Algorithm
Let $R_0 := [0,1]^{m.n}$ and ε be a sufficiently small positive number. Set *restart := true*;
Solve the convex problem (15) with $K \leftarrow K_{R_0} = \Omega \cap R_0$ to obtain a solution z^{R_0} and the first lower bound $\beta_0 := \beta(R_0)$;
If z^{R_0} is feasible to (11) **then**
set $\gamma_0 := f(z^{R_0})$, $z^0 := z^{R_0}$, *restart := false* **else** $\gamma_0 := +\infty$;
Endif
If $(\gamma_0 - \beta_0) \le \varepsilon|\gamma_0|)$ **then** STOP, z^0 is an ε-optimal solution of (11) **else** set $\Re \leftarrow \{R_0\}$, $k \leftarrow 0$;
Endif
While (STOP = false) **do**
Select a rectangle R_k such that $\beta_k = \beta(R_k) = \min\{\beta(R) : R \in \Re\}$.
Let $j^* \in \{1, \ldots, m.n\}$ be the index such that $z_{j^*}^{R_k} \notin \{0, 1\}$. Divide R_k into two sub-rectangles R_{k_0} and R_{k_1} via the index j^*:

$$R_{k_i} = \{z \in R_k : z_{j^*} = i; i = 0, 1\}. \tag{18}$$

Solve the subproblems (P_{k_i}) to obtain $\beta(R_{k_i})$ and $(z^{R_{k_i}})$:

$$(P_{k_i}) \quad \beta(R_{k_i}) = \min\{f(z) : z \in \Omega, z \in R_{k_i}\}. \tag{19}$$

For $i = 0, 1$
  If $z^{R_{k_i}}$ is feasible to (11) **then**
    update γ_k and the best feasible solution z^k;
  Endif
Endfor
Set $\Re \leftarrow \Re \cup \{R_{k_i} : \beta(R_{k_i}) < \gamma_k - \varepsilon, i = 0, 1\} \setminus R_k$.
If $\Re = \emptyset$ **then** STOP, z^k is an ε-optimal solution **else** set $k \leftarrow k + 1$.
Endwhile

5 Numerical results

The algorithms are written in language C on Microsoft Visual C++ 2008. The implementation takes place on a notebook with chipset Intel(R) Core(TM) Duo CPU 2.0 GHz, RAM 3GB. The commercial software CPLEX 11.2 is used as a convex quadratic programming solver.

The following notations are used in these tables:

- Pb: Problem,
- Bin: number of the binary variables,
- $Ctrs$: number of the constraints,
- Obj: value of the objective function obtained by each algorithm,
- $Time$: CPU time in seconds of each algorithm,
- $Gap\% = \frac{Obj - Lastlowerbound}{Obj}$,
- $GapCE\% = \frac{Obj - Firstlowerbound}{Obj}$.

In Table 1, we give a comparison between CE and B&B. The results have demonstrated that with small dimension, CE gives good solutions as the same as B&B does, but CPU time in our approach is very better. The ratio of time consumed varies from 114 to 10,218. Especially, for Problem 4, B&B runs for more than one hour and does not produce a solution, whereas CE gives a good solution. In this problem, we compute GapCE by using the first lower bound. Figures 1 and 2 show the result for nine out of ten problems. In Table 2, we continue to compare in larger dimension, with $m = 20, n = 10$. We can see that CE still works very well while B&B runs for more than one hour and does not produce a solution, for all problems. In these experiments CE was run with number of samples, $N = 50$, and $\theta = 0.4$. The number of iterations is limited to 20.

Table 3 gives the the results with 10 instances with large dimensions having $50 \times 30 = 1{,}500$ binary variables. The weights $\{w_j\}$ are random integers chosen uniformly from 1 to 10, and the task success probabilities $\{r_j\}$ are random numbers chosen uniformly in interval $[0, 1]$. The parameters are chosen as follows: number of samples is $N = 100$, $\theta = 0.4$. Table 4 presents the results for very large dimensions. In this case we take $\theta = 0.04$. The maximum of number of iterations is 50.

Table 1 CE compares with B&B, $m = 10$, $n = 10$

Data			CE method			B&B		
Pb	Bin	Ctrs	Obj	Time(s)	Gap	Obj	Time(s)	Gap
1	100	10	29.425828	0.078	0.1999	29.425858	170.890	0.2
2	100	10	40.899408	1.156	0.5556	40.790448	1919.812	0.29
3	100	10	30.256247	0.078	1.2553	29.927309	776.766	0.17
4	100	10	26.001069	1.140	2.85	NA	>1 h	NA
5	100	10	34.964398	0.078	0.06	34.964398	8.937	0.06
6	100	10	28.286803	2.890	1.03	28.286803	2887.265	1.03
7	100	10	35.185966	0.078	0.3965	35.454166	47.859	1.15
8	100	10	29.528116	0.078	0.33	29.528116	23.110	0.33
9	100	10	19.188199	0.078	0.46	19.188199	38.578	0.46
10	100	10	15.761578	0.078	2.08	15.761578	797.063	2.08

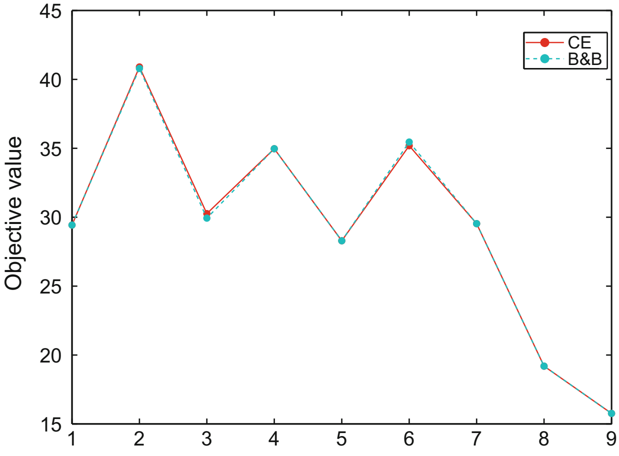

Fig. 1 Objective function, $m = 10$, $n = 10$

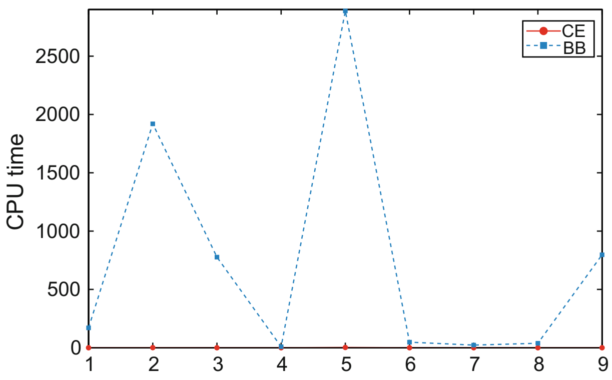

Fig. 2 CPU time, $m = 10$, $n = 10$

Table 2 CE compares with B&B, $m = 20, n = 10$

Data			CE method			B&B	
Pb	Bin	Ctrs	Obj	Time(s)	GapCE	Obj	Time (h)
1	200	20	23.765072	0.203	0.2	NA	>1
2	200	20	21.351345	0.218	0.56	NA	>1
3	200	20	22.799311	0.187	1.26	NA	>1
4	200	20	23.388861	0.203	2.85	NA	>1
5	200	20	18.416153	0.187	0.06	NA	>1
6	200	20	25.096867	0.203	1.03	NA	>1
7	200	20	22.698760	0.187	0.4	NA	>1
8	200	20	19.459113	0.218	0.33	NA	>1
9	200	20	13.077463	0.187	0.46	NA	>1
10	200	20	23.539310	0.203	2.08	NA	>1

Table 3 Results with large dimensions: $m = 50, n = 30$

Instance	Bin	Ctrs	Time(s)	GapCE
1	1,500	50	4.766	2.562583
2	1,500	50	4.750	3.315747
3	1,500	50	4.797	2.816054
4	1,500	50	4.797	1.868424
5	1,500	50	5.047	3.784732
6	1,500	50	4.875	2.093580
7	1,500	50	4.968	3.010455
8	1,500	50	4.797	1.401817
9	1,500	50	4.937	3.711038
10	1,500	50	4.860	2.658826

Table 4 Results with very large dimensions

Instance	m	n	Bin	Ctrs	Time (s)	GapCE	Samples
1	300	100	30,000	300	141.078	1.270807	1,000
2	500	100	50,000	500	229.593	1.829299	1,000
3	500	300	150,000	500	1190.437	3.139713	2,000
4	700	500	350,000	700	2306.141	4.031592	2,000
5	1,000	500	500,000	1,000	3649.781	5.114315	2,000

6 Conclusion

In this work, we have firstly proposed an approach based on the CE method for solving UAV Task Assignment problem, and then presented a global approach based on B&B algorithm for measuring the quality of our CE algorithm. The results have shown the efficiency of this approach not only with small dimensions but also with very large dimensions. This approach can overcome very well the barrier of binary variables, that the standard methods are usually very difficult for treating. Other non-linear models of task assignment problem will be studied in our future work.

References

1. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to DC programming: theory, algorithms and applications (dedicated to Professor Hoang Tuy on the occasion of his 70th birthday). Acta Math. Vietnamica **22**, 289–355 (1997)
2. Le Thi, H.A., Pham Dinh, T.: Large scale global molecular optimization from exact distance matrices by a DC optimization approach. SIAM J. Optim. **14**(1), 77–114 (2003)
3. Le Thi, H.A., Pham Dinh, T.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Ann. Oper. Res. **133**, 23–46 (2005)
4. Chandler, P.R., Pachter, M., Rasmussen, S.R., Schumacher, C.: Multiple Task Assignment for a UAV Team. AIAA Guidance, navigation, and control conference, Monterey, Aug. (2002)
5. Costa, A., Dafydd, O., Kroese, D.: Convergence properties of the cross-entropy method for discrete optimization. Oper. Res. Lett. **35**(5), 57380 (2007)
6. Cruz, J.B. Jr., Chen, G., Li D., Wang, X.: Particle swarm optimization for resource allocation in UAV cooperative control. AIAA Guidance, Navigation, and Control Conference, Providence, Aug. (2004)
7. Fontes, D.B.M.M., Fontes, F.A.C.C.: Minimal switching time formations with collision avoidance. Dyn. Inf. Syst. Springer Optim. Appl. **40**, 305–321 (2010)
8. Dambreville, F.: Cross-entropy method: convergence issues for extended implementation. In: Proceedings of the Rare Event Simulation Conference (RESIM 2006), Bamberg, Germany (2006)
9. Walker, D.H., McLain, T.W., Howlett, J.K. : Coordinated UAV target assignment using distributed tour calculation. In: Grundel, D., Murphy, R., Pardalos, P.M. (eds.) Theory and Algorithms for Cooperative Systems, Series on Computers and Operations Research, vol. 4, pp. 327–333. Kluwer, Dordrecht (2004)
10. Dubin, U.: The cross-entropy method for combinatorial optimization with applications. Master's thesis, The Technion, Israel Institute of Technology, Haifa, June (2002)
11. Dubin, U.: Application of the cross-entropy method for image segmentation. Ann. Oper. Res. (2004) (submitted)
12. Pasiliao, E.L.: Local neighborhoods for the multidimensional assignment problem. In: Hirsch, M.J., Pardalos, P.M., Murphey, R. (eds.) Dynamics of Information Systems, Springer Optimization and its Applications, vol. 40, pp. 353–371. Springer, Berlin (2010)
13. Krokhmal, P., Murphey, R., Pardalos, P., Uryasev, S. : Use of conditional value-at-risk in stochastic programs with poorly defined distributions. In: Butenko, S., Murphey, R., Pardalos, P. (eds.) Recent Developments in Cooperative Control and Optimization, pp. 225–242. Kluwer, Dordrecht (2004)
14. Le Ny, J., Feron, E.: An Approximation Algorithm for the Curvature-Constrained Travelling Salesman Problem. 43rd Annual Allerton Conference on Communications, Control and Computing, Monticello, Sept. (2005)
15. Murray, R.M.: Recent research in cooperative control of multivehicle systems. J. Dyn. Syst. Measure. Control **129**(5), 571–583 (2007)
16. Nygard, K.E., Altenburg, K., Tang, K., Schesvold, D.: A decentralized swarm approach to asset patrolling with unmanned air vehicles. In: Grundel, D., Murphy, R., Pardalos, P.M. (eds.) Theory and Algorithms for Cooperative Systems, Series on Computers and Operations Research, vol. 4, pp. 327–338. Kluwer, Dordrecht (2004)
17. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs (1982)
18. Pardalos, P.M., Pitsoulis, L.S. (eds.): Nonlinear Assignment Problems: Algorithms and Applications. Combinatorial Optimization, vol. 7. Kluwer, Dordrecht (2000)
19. Protvin, J.Y.: Genetic algorithms for the travelling salesman problem. Ann. Oper. Res. **63**(3), 339–370 (1996)
20. Richards, A., Bellingham, J., Tillerson, M., How, J.P.: Coordination and control of multiple UAVs. AIAA Guidance, Navigation and Control Conference, Monterey, Aug. (2002)
21. Richards, A., How, J.P.: Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Progrmming. American Control Conference, Anchorage, AK, May (2002)

22. Rubinstein, R.Y.: Optimization of cumputer simulation models with rare events. Eur. J. Oper. Res. **99**, 89–112 (1997)
23. Rubinstein, R.Y.: The simulated entropy method for combinatorial and continuous optimization. Methodol. Comput. Appl. Probab. **2**, 127–190 (1999)
24. Rubinstein, R.Y.: Combinatorial optimization, cross-entropy, ants and rare events. In: Uryasev, S., Pardalos, P.M. (eds.) Stochastic Optimization: Algorithms and Application, pp. 304–358. Kluwer, Dordrecht (2001)
25. Rubinstein, R.Y.: The cross-entropy method and rare-events for maximal cut and bipartition problems. ACM Trans. Model. Comput. Simul. **12**(1), 27–53 (2002)
26. Rubinstein, R.Y., Kroese, D.P.: The Cross-Entropy Method: A Unified Approach to Monte Carlo Simulation. Randomized Optimization and Machine Learning. Springer, Berlin (2004)
27. Salman, A., Ahmad, I., Al-Madani, S.: Particle swarm optimization for task assignment problem. Microprocess. Microsyst. **26**(8), 363–371 (2002)
28. Schumacher, C., Chandler, P., Pachter, M., Pachter, L.: Constrained Optimization for UAV Task Assignmnet. AIAA Guidance, Navigation, and Control Conference, Providence, Aug. (2004)
29. Maddula, T., Minai, A.A., Polycarpou, M. : Multi-target assignment and path planning for groups of UAVs. In: Butenko, S., Murphey, R., Pardalos, P. (eds.) Cooperative Control and Optimization, pp. 261–272. Kluwer, Dordrecht (2004)
30. Jin, Y., Polycarpou, M., Minai, A.A.: Cooperative real-time task allocation among groups of UAVs. In: Butenko, S., Murphey, R., Pardalos, P. (eds.) Cooperative Control and Optimization, pp. 207–224. Kluwer, Dordrecht (2004)