# Artificial Intelligence Methods (COMP2051 or AE2AIM)

Prof. Ruibin Bai Spring 2019

## Coursework: Memetic Algorithm for Multi-Knapsack Problem

### 1. Introduction

Multi-dimensional knapsack problem is a classic NP-Hard combinatorial optimisition problem used to test the performance metaheuristics. In this coursework, you are asked to write a C program to solve this problem using a **memetic algorithm** (a variant of genetic algorithm). In addition to submitting source code, a report (no more than 2000 words and 6 pages) is required to describe the algorithm, the experimental results, discussions and reflections on results and performance of the algorithm. **This coursework carries 50% of the module marks**. The rest of module marks comes from the final written exam.

### 2. Multi-dimensional Knapsack Problem

Multi-dimensional knapsack problem is an extension of the 1D knapsack problem by adding capacity constraints in multiple dimensions. The problem can be formally defined as follows. Given a set of $n$ items numbered from 1 up to $n$, each with a size vector $\mathbf{v}=(v_{1j}, v_{2j}, v_{3j}, \ldots, v_{mj})$ where $v_{ij}$ is the $i$-th dimensional size of item $j$. $b_i$ is the $i$-th dimensional size of knapsack. $p_j$ is the profit of item $j$ if it is included in the knapsack. Denote $x_j$ be the binary variables to indicate whether item j is included in the knapsack (=1) or not (=0). The problem to be solved is then formulated as follows

$$\sum_{j=1}^{n} p_j x_j$$

Subject to:

$$\sum_{j=1}^{n} v_{ij} x_j \leq b_i \qquad i = 1, \ldots, m$$

$$x_j = \{0,1\}$$

3.

## 3. Problem instances

In this lab, you are asked to attempt some more challenging instances from paper: P.C. Chu and J.E.Beasley "A genetic algorithm for the multidimensional knapsack problem", Journal of Heuristics, vol. 4, 1998, pp63-86. All the data files are compressed in `mknap-instances.zip`, downloadable from Moodle. The zip file includes 9 problem instance files (each containing 30 instances), 1 data format file `file-format.txt` and 1best known solution file `best-feasible-slns.txt`.

## 4. Experiments conditions and requirements

Your algorithms are expected to stop after a predefined computational time (e.g. 5min). *More details shall be given later*.

## 5. Marking criteria

- The quality of the experimental results (30%).
- The quality of codes (30%)
- Report (40%)

## 6. Submission deadline

**29th April 2019, 4pm Beijing Time**

## 7. How to submit

*To be confirmed*.