**Discrete Optimization**

Elizabeth Davenport, Megan Johnston, Calli Nguyen, Shuowen Wei

Apr. 18

1. An general introduction to discrete optimization:

%%%%

Citation: 5

What’s discrete optimization? Well, discrete optimization is a very important part of optimization that mainly in applied mathematics and computer science area. As opposed to continuous optimization, the variables used in the some or all of the mathematical problems are restricted to discrete values, such as the integers. Two notable branches of discrete optimization are:

* combinatorial optimization, which refers to problems on graphs, matroids and other discrete structures
* integer programming

citation 5

%%%%

Combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects. [7]

An integer programming problem is a mathematical optimization in which some or all of the variables are restricted to be integers. [8]

Some examples in discrete optimization are as follows:

* Integer linear programming
* Set cover problem
* Knapsack problem
* Graph theory
  + Minimum spanning tree
  + Vertex cover problem
  + Traveling salesman problem (Hamiltonian circuit)
  + Shortest path problem
* Scheduling problem
  + Maximum flow problem

%%%%

Citation 8

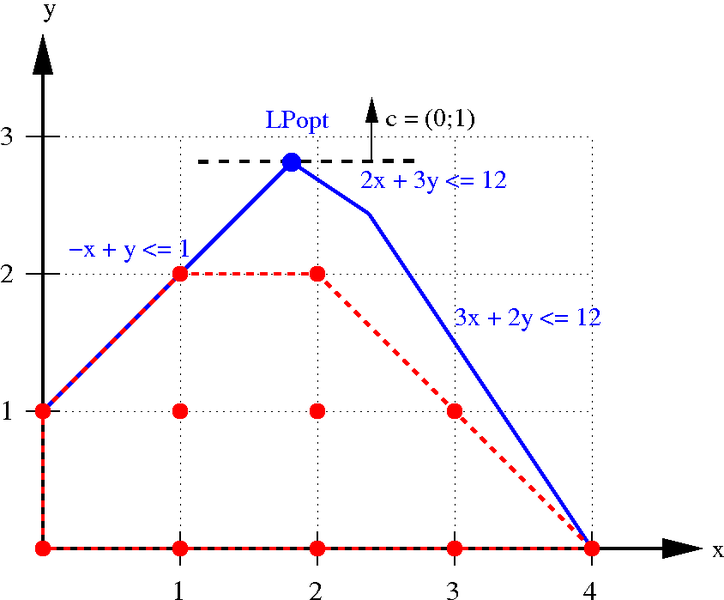
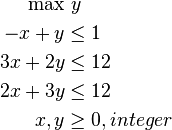
An integer linear programming problem can be typically expressed as follows:

maximize

subject to

and integer

An example of ILP:



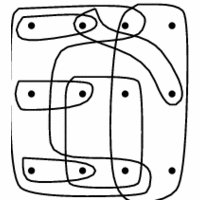
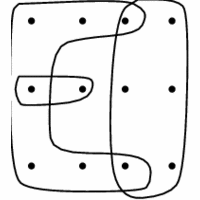
Citation 8

%%%%

%%%%

Citation 6

The set cover problems is a NP-hard problem, this problem is state as: given a set of elements "{1, 2, … , n}" (called the universe) and a set 𝑆 of 𝑚 sets with their union equals the whole universe {1, 2, …, n}, then the set cover problem is to find the smallest subset of  the union of which contains all elements in the universe.

While we should point out that the exact set cover problem is NP-complete, and the two graphs above are not exact set cover.

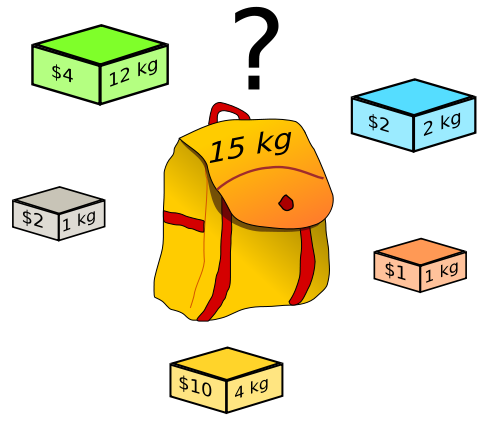
Citation 6

%%%%

%%%%

Citation 9

Knapsack problems is a also NP-hard problem, this problem is described as: given a set of items, each with a weight and a value , so then Knapsack problem is asked to determine the number of each item to collect so that the total weight is less than or equal to a given limit capability of the “ bag” and meanwhile the total value is as large as possible：



We also should point out that the 0-1 knapsack problem is NP-complete, we can prove that it is polynomial mapping reducible from exact set cover problem, which is also NP-complete as we mentioned above.

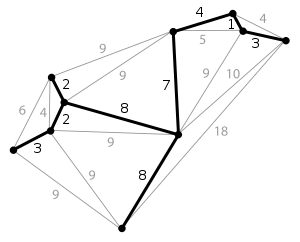
Citation 9

%%%%

%%%%

Citation 10

A minimum spanning tree (MST) problem is that, given a (usually undirected) graph, then to find a spanning tree hose weight is less than or equal to the weight of every other spanning tree. This problem can be solved by Kruskal's algorithm (which is a very famous greedy algorithm), with steps where is the number of vertices of the input graph. An example is as follows:



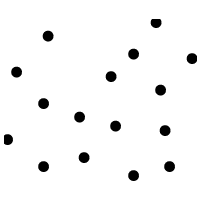
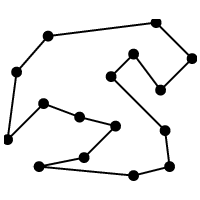
Citation 10

%%%%

%%%%

Citation 6

The traveling salesman problem is also NP-hard, whichasks the following question: given a bunch of cities and the distances between each pair of cities, can the salesman come up with a shortest possible route that visits each city exactly once and finally returns to the origin city?

Citation 6

%%%%

%%%%

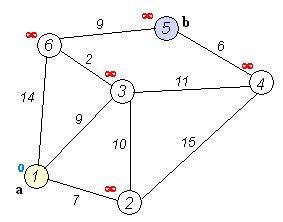
Citation 11

In graph theory, the shortest path problem in a ( usually undirected) graph is that, to find a path between one or all pairs of vertices in the graph such that the sum of the vaues/distances of all the edges in that path is minimal:

* The single-source shortest path problem is to find the shortest paths from a source vertex v to all other vertices in the given graph.
* The single-destination shortest path problem is to find shortest paths from all vertices in the **directed** graph to a single destination vertex v. This can obviously be reduced to the single-source shortest path problem by reversing the arcs in the directed graph.
* The all-pairs shortest path problem is to find shortest paths between every pair of vertices v, v' in the graph.

Dijkstra's algorithm can solve the single-source shortest path problems in steps where n is the number of vertices in the graph.

[Floyd–Warshall algorithm](http://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm) can solve all pairs shortest paths in steps where n is the number of vertices in the graph.



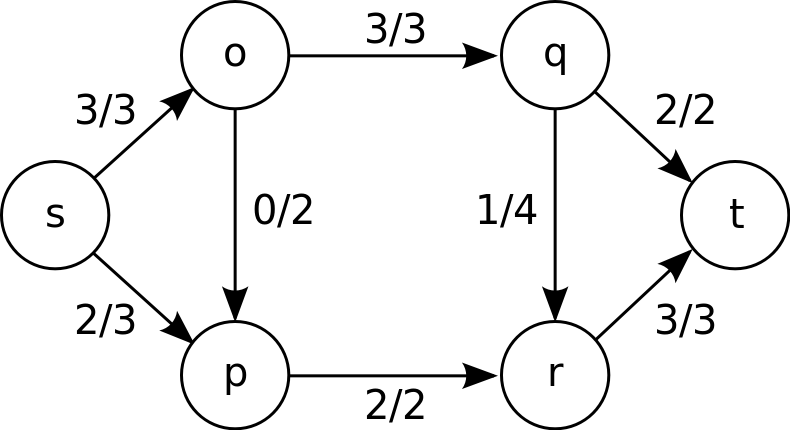
Citation 11

%%%%

%%%%

Citation 12

The maximum flow problem is to find a feasible flow through a single-source (s), single-sink (t) flow network that is maximal, and the max-flow min-cut theorem states that the maximum value of an s-t flow is equal to the minimum capacity over all s-t cuts:



Citation 12

%%%%

2. Discrete Optimization in Medical Imaging:

In order to compare two medical images in the same space they must be registered to each other, or to a standard template. This allows for the same point to correspond to the same anatomical feature in both images. Automated and accurate registrations are crucial for group comparisons when multiple images must be compared. Medical image registration is faced with four main obstacles: curse of dimensionality, curse of non-convexity, curse of non-linearity, and the curse of modularity[1](#_ENREF_1). The curse of dimensionality occurs when more parameters must be added to the system to increase the powerfulness of the model, but this also increases the complexity of the problem. This is often seen in registration procedures that require high dimensional warping. The curse of non-convexity occurs when there are more parameters input than constraints, or, in other words, it is an ill-posed problem. This often occurs when an image of many voxels is being related to a single bio-marker. The curse of non-linearity occurs because the parameters are related in a highly non-linear fashion. This is again seen in registration procedures with high dimensional warping. The curse of modularity occurs because solutions are found for very specific problems, rather than a broad application. Different imaging modalities, CT or MRI, usually need their own registration algorithm due to differences in the intensity values. MRI is specifically difficult because the intensity values are not consistent across scanners, where a CT image will have a standard unit of measure. This makes MRI registration especially difficult because it cannot be based on exact intensities, but on the differences, and geometry in the image.

These problems make image registration, and other imaging problems, a prime candidate for the implementation of optimization algorithms. Most image registration algorithms require an iterative approach. At each iteration the similarity between the two “registered” images is calculated, and the algorithm continues moving towards a better similarity measure, until it converges within some tolerance. Many optimization methods have been used for image registration in the past. These methods include: Powell’s (conjugate direction) method, Downhill Simplex, Levenberg-Marquardt, Newton-Raphson, stochastic search methods, gradient descent methods, and quasi-exhaustive search methods[2](#_ENREF_2). Recently, Glocker et al. published a registration algorithm using linear programming[3](#_ENREF_3). Linear programming is a branch of combinatorial optimization, within discrete optimization.

Glocker et al.’s methods modeled the registration as a Markov Random Field (MRF), where a set of labels is associated with a set of deformations. The MRF is then optimized through a method known as Fast-PD[4](#_ENREF_4). This method uses the primal-dual schema. The primal-dual schema, first starts by seeking an optimal solution x\* to an integer program (the primal problem), which is NP-hard. The problem is then relaxed to get a primal (minimization) and a dual (maximization) linear program. The pair of linear programs are shown below:

View the MathML source

From this the main goal is to minimize gap between primal and dual. This is illustrated below:

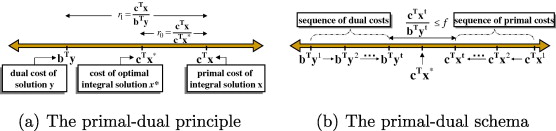


Figure 1: Dual and Primal solutions make local improvements to each other until their final costs (bTyt, cTxt) are within a pre-set range

This method was tested on the automatic registration of individual brain MRI images to a template, and it was successful. Particularly, it addressed existing imaging issues such as the curse of non-convexity by implementing the primal-dual optimization technique.

References:

1. Paragios N. Graphical Models and Discrete Optimization in Biomedical Imaging: Theory and Applications. *Seminars*: Isaac Newton Institue for Mathematical Sciences; 2011

2. Maintz JBA, Viergever MA. An overview of medical image registration methods. *UU-CS* 1998

3. Glocker B, Komodakis N, Tziritas G, et al. Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis* 2008;12:731-741

4. Komodakis N, Tziritas G. Approximate labeling via graph cuts based on linear programming. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2007;29:1436-1453

5. <http://en.wikipedia.org/wiki/Discrete_optimization>

6. <http://www.cs.sunysb.edu/~algorith/implement/syslo/implement.shtml>

7. <http://en.wikipedia.org/wiki/Combinatorial_optimization>

8. <http://en.wikipedia.org/wiki/Integer_programming>

9. <http://en.wikipedia.org/wiki/Knapsack_problem>

10. <http://en.wikipedia.org/wiki/Minimum_spanning_tree>

11. <http://en.wikipedia.org/wiki/Shortest_path_problem>

12. <http://en.wikipedia.org/wiki/Maximum_flow_problem>