

Project Definition

An *optimization problem* is a problem in which we have a collection of decisions to make, a set of rules governing how these decisions interact, and a means of quantitatively comparing possible solutions. Our goal in such a problem is to find the optimal set of decisions that collectively satisfy all of the rules.

We encounter optimization problems in all aspects of our lives, from the most mundane to the most critical decisions that we face. In some cases, we can easily analyze the situation to make good decisions. However, many optimization problems are too complex to solve simply by observation or using rules-of-thumb. An alternative approach is to represent the optimization problem as a mathematical model, often called a *mathematical program*, in which each of the decisions is represented by a variable, each rule is enforced by a mathematical constraint, and an objective function captures the goal of finding the best solution. Mathematical programming has been successful in solving an enormous class of real-world problems. Uses range from determining the structure of a transportation system to constructing a financial portfolio to determining the most effective way to orient beams when administering radiation to a cancer patient.

Large-scale optimization problems are optimization problems in which the number of variables and/or constraints is exponentially large. For example, a *crew scheduling problem*, solved routinely by major airlines in order to assign pilots to flights, typically has on the order of *billions* of decision variables. It is interesting to note that the underlying real-world situation often does not have such a large number of variables or constraints. Instead, the size is a function of the mathematical model chosen to represent the real-world problem. Ironically, it is sometimes necessary to make “small” real-world problems very large in order to attain a structure that is mathematically tractable.

Many real-world problems requiring the use of optimization techniques can be modeled as a *network* – a series of *nodes* and *arcs*. For example, nodes might represent warehouses and retailers, while arcs might correspond to the movement of finished goods between these facilities. *Network flow problems* – problems in which we decide how to move materials through a defined network – have been well studied, and techniques exist for effectively solving many such problems.

Less well studied than network flow problems, and usually far more challenging, are *network design problems*. These are problems in which we not only need to decide the most efficient way to move materials through a network, but the *structure* of the network as well – which nodes and arcs to include, and how much capacity to assign to them. Network design problems occur in a wide number of areas, including but not limited to transportation, supply chain logistics, and telecommunications. The mathematical network might correspond quite closely to a physical network – for example, arcs might represent stretches of roadway to be built and nodes might represent physical intersections. However, a mathematical network might also be an abstraction that has a less obvious physical connection. Arcs often represent movements not only in space, but in time as well, and nodes frequently represent events rather than physical locations.

When solving an optimization problem, the tractability of the mathematical model depends on certain structural characteristics. For example, most models with continuous variables, linear constraints, and a linear objective function are straightforward to solve –

even when such problems are very large. If the constraints and objective function are linear but the variables are discrete (for instance, variables representing the number of people to include in a group must be limited to only take integer values), then the problem becomes more difficult to solve, and the size of the model takes on greater importance. When non-linearities are introduced, either into constraints or the objective function, problems often become far more challenging and in many cases exceed current algorithmic capabilities. Such problems are often solved heuristically, with no proof of optimality and often not even a bound to estimate the quality of the solution.

Given the limited tools available for solving non-linear models, another approach is to develop an alternative formulation in which the non-linearities have been removed. This linearization of the problem frequently comes at the cost of a drastic increase in problem size (the previously mentioned crew scheduling example has an exponential number of variables as a result of removing a non-linear objective function, for example). Thus, the challenge of solving a non-linear mathematical program is replaced with the challenge of solving a very large linear program, typically with integer variables.

Network design problems often are plagued with both integer variables *and* non-linearities, particularly in the cost function. For example, a problem might contain a candidate set of nodes such as potential warehouse sites or possible times for an event to occur. For each candidate node, a *binary variable* might be assigned. This variable is restricted to values 1 (corresponding to the decision of including the node in the network) and 0 (not including the node). In addition to this set of binary variables, network design problems often have non-linear cost functions. Consider, for example, the cost associated with flowing material over an arc. In some cases, this cost is linear – there is a fixed per-unit cost, and we obtain the cost of a solution by simply multiplying the total flow on the arc by the cost per unit. However, this is often not the case. For example, consider a trucking company moving freight across the country. Whether they have a truck that is 25% full, 50% full, or 100% full, they still have to pay the driver the same amount. Thus, the cost function is not linear but instead a *step function* – all flows from 0 to 1 have the same cost; flows from 1 to 2 have twice this cost, and so on.

Our objective is to develop new methods for solving network design problems with non-linear costs. Our focus is on doing so by *developing alternative models* with linear constraints and objective function (but very large numbers of discrete variables), and by *developing algorithmic techniques* to make these problems tractable.