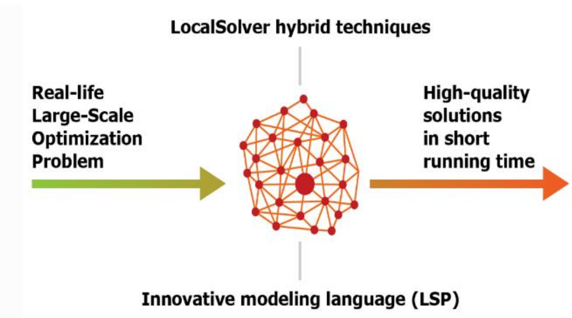
LocalSolver

> **Large-scale food supply chain optimization at Pasco Shikishima**
>
> *We developed the Pasco optimization model using LocalSolver within a few days of work. LocalSolver provides outstanding solutions, as considered by Pasco planners, in just a few minutes of running times, while the number of variables is gigantic (tens of millions). We could not imagine that such a feat was possible at first, since state-of-the-art MIP solvers like CPLEX, Xpress and Gurobi were unable to tackle the problem in hours. Now we know that using LocalSolver, it's possible!*
>
> Mr. Shinichi Kuroda,
> Project director

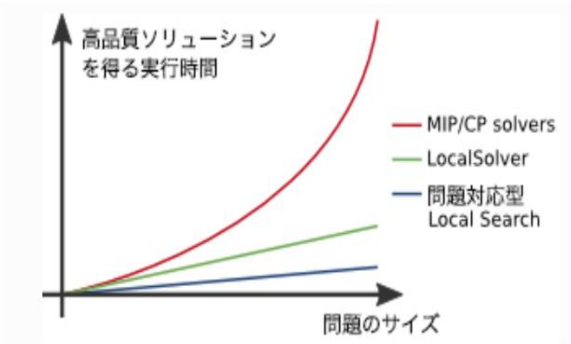
All-terrain & all-in-one optimization solver

Having modeled your optimization problem using natural mathematical constructs, LocalSolver provides you with high-quality solutions in short running times. Based on an innovative resolution technology, LocalSolver scales up to millions of variables, running on basic computers. LocalSolver includes a high-level math modeling language for fast prototyping and lightweight object-oriented APIs for tight integration, which makes it easy to use and deploy on any platform.



- For discrete, numerical, black-box problems
- Fast solutions to million-variable problems
- High-level math modeling language
- Innovative resolution technology
- Easy APIs for Python, C++, Java, C#
- Simple and transparent licensing
- Dedicated and responsive support
- Free for teaching and pure research

Discover LocalSolver unique set-based modeling features for routing & scheduling problems




```
// Capacited Vehicle Routing Problem (CVRP)
function model() {
   // for each vehicle, the list of visited clients
   routes[1..nbTrucks] <- list(nbCustomers);
   constraint partition  [k in 1..nbTrucks](routes[k]);

   for[k in 1..nbTrucks] {
      route <- routes[k];
      c <- count(route);
      // truck capacity constraint
      constraint sum(0..n-1, i => demands[route[i]]) <= capacity;
       // sum of distances between each pair of clients and depot
      distances[k] <- sum(0..n-2, i => distance(route[i], route[i+1]))
          + distance(depot, route[0]) + distance(route[n-1], depot);
   }
   minimize sum[k in 1..nbTrucks](distance[k]);
}
```

LocalSolver に関するお問合せ

MSI 株式会社(日本配給元)
〒261-7102
千葉市美浜区中瀬 2-6 WBG2 階
Tel:043-297-8841　fax:043-297-8836
Email:localsolver@msi-jp.com
Web:www.msi-jp.com/localsolver/

Why LocalSolver?

LocalSolver combines several optimization techniques to solve your problems at hand. Relying on an innovative hybrid resolution technology, it delivers high-quality solutions in short running times, while being able to scale to million-variable problem.

All-terrain & all-in-one

Models supported by LocalSolver involve linear and nonlinear constraints and objectives including algebraic, logical, set and black box expressions, in continuous and discrete variables. LocalSolver provides high-quality solutions in seconds to million-variable problems. The resolution is direct: no need of complex tuning. The licencing and pricing are simple and transparent.

Innovative math modeling language

LocalSolver comes with a powerful modeling and scripting language. It enables you to quickly prototype optimization applications. The LocalSolver Programming language (LSP) offers an efficient programming style: dynamic but strongly typing, implicit variable declaration, compact looping syntax, etc. .
Our goal is to reduce your programming effort as much as possible, while framing your prototyping work. You will see that the resulting LSP models are less verbose and more readable than the ones written with existing modeling languages.

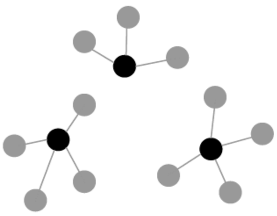
Easy object-oriented APIs

To integrate LocalSolver in your business applications, we provide easy-to-use object-oriented programming interfaces (APIs) for Python, Java, C# and C++. LocalSolver's APIs are lightweight, with only a few classes. Passing from LSP to APIs is easy: you have to concentrate on your math optimization model only.

Reliable, robust and fully portable

Because efficiency is nothing without reliability and robustness, we ensure our clients the best quality thanks to a drastic continuous integration methodology, coupled with a responsive and dedicated support realized by PhD-level optimization experts. LocalSolver is fully portable : it is available on Windows, Linux, macOS and Solaris, for 32 ands 64 bits architecture.

Some examples

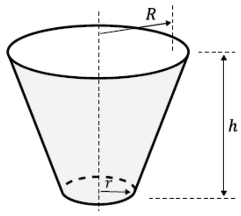


Facility location

```
// Select a subset P among N points minimizing the sum of distances
// to each point from N to the nearest point in P.
function model() {
   // decisions: point i is in P iif x[i] = 1
   x[1..N] <- bool();

   // Select P points among N
   constraint sum[i in 1..N](x[i]) == P;
   minDist[i in 1..N] <- min[j in 1..N](x[j] ? Dist[i][j] : +inf);
   minimize sum[i in 1..N](minDist[i]);
}
```

Parametric optimization



$$V = \frac{\pi h}{3}(R^2 + Rr + r^2)$$

$$S = \pi r^2 + \pi(R + r)\sqrt{(R - r)^2 + h^2}$$

```
// Maximize the volume of a bucket with a given surface of metal
function model() {
   R <- float(0,1);
   r  <- float(0,1);
   h <- float(0,1);
   V <- PI * h/3.0 * (R * R + R * r + r * r);
   S <- PI * r * r + PI * (R + r) * sqrt(pow(R-r,2) + h * h);
   constraint S <= SMAX;
   maximize V;
}
```

Some clients

VEOLIA · EDF · ENGIE · FUJITSU · dynacom · PSA GROUPE · BOSCH · RENAULT · SIEMENS · AIR LIQUIDE · AIRBUS · AMADEUS · SITA · JCDecaux · SNCF