Savings, Savings, Savings

by

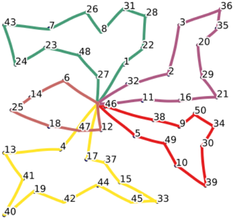
Gabriel C-Parent

Overview

- cvrp problem
- implementation details
- improvement procedures
- construction procedure
- genetic algorithm
- tabu search
- QA

The Problem

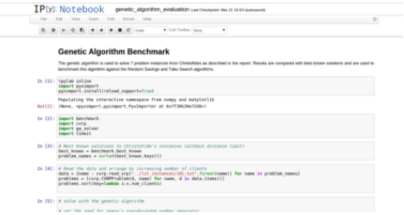
Capacitated Vehicle Routing



$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{route \in solution} distance(route) \\ \text{subject to} & weight(route) \leq vehicle\ capacity \end{array}$$

Implementation

Implementation Details


IP[y]: Notebook
genetic_algorithm_evaluation Last Checkpoint: Mar 21 10:54 (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Code
Cell Toolbar:
None

Genetic Algorithm Benchmark

The genetic algorithm is used to solve 7 problem instances from Christofides as described in the report. Results are compared with best known solutions and are used to benchmark this algorithm against the Random Savings and Tabu Search algorithms.

```
In [1]: %pylab inline
        import pyximport
        pyximport.install(reload_support=True)

        Populating the interactive namespace from numpy and matplotlib
Out[1]: (None, <pyximport.pyximport.PyxImporter at 0x7f30b39e72d0>)

In [2]: import benchmark
        import cvrp
        import ga_solver
        import timeit

In [3]: # Best known solutions to Christofide's instances (without distance limit)
        best_known = benchmark.best_known
        problem_names = sorted(best_known.keys())

In [4]: # Read the data and arrange by increasing number of clients
        data = {name : cvrp.read_vrp("../lit_instances/{0}.txt".format(name)) for name in problem_names}
        problems = [cvrp.CVRPProblem(d, name) for name, d in data.items()]
        problems.sort(key=lambda x:x.num_clients)

In [5]: # solve with the genetic algorithm

        # set the seed for numpy's pseudorandom number generator
```

NumPy

Cython

Implementation Details

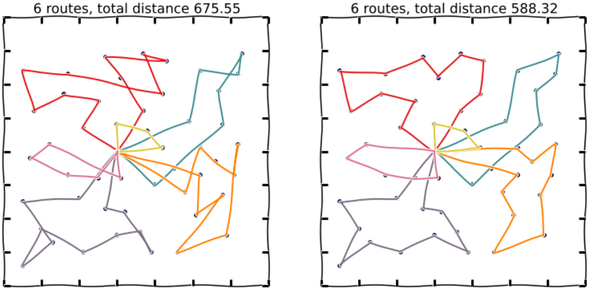
- reuse basic operators
- modularity
- concise

Improvement

2-opt descent

- uses common 2-opt operator
- calculates all possible 2-opt for each iteration
- chooses the best available

2-opt example



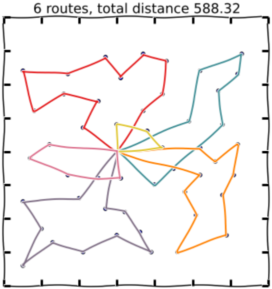
6 routes, total distance 675.55

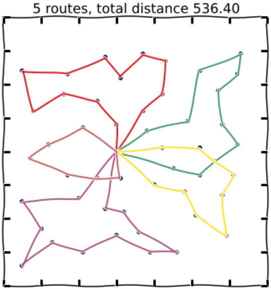
6 routes, total distance 588.32

λ_1-interchange definition

- λ-interchange, Osman, 1991
- exchange of customers between routes
- only feasible exchanges (capacity constraint)
- insertion (1, 0) and (0, 1) or interchange (1, 1)
- chooses the best option at each iteration
- apply 2-opt descent on routes implicated

λ_1-interchange example



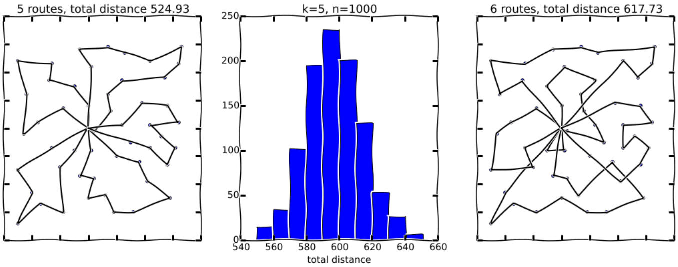


Construction

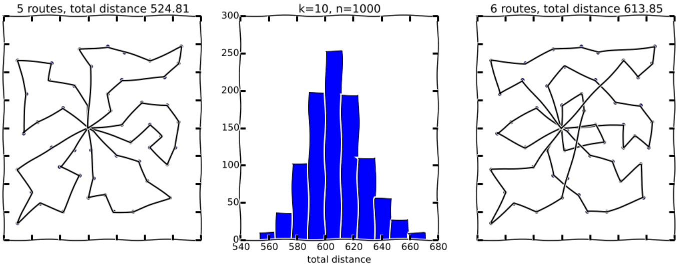
Random Savings Definition

- iterated local search
- variant of parallel savings
- at each iteration select randomly from top k best savings
- k=1 → normal parallel savings
- once finished, apply improvement method

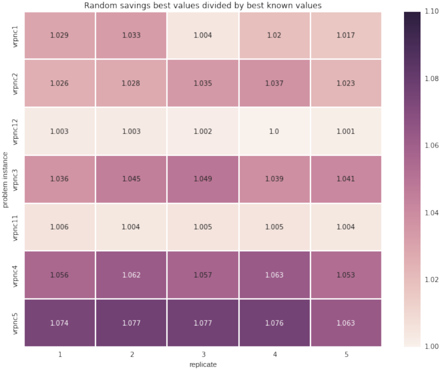
Random Savings, k=5



Random Savings, k=10



Best result in 60 secs

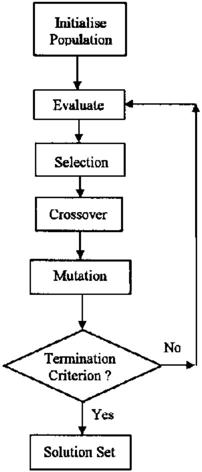


Random savings best values divided by best known values

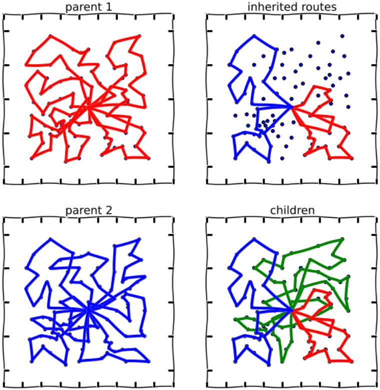
problem instance	1	2	3	4	5
vrpnc1	1.029	1.033	1.004	1.02	1.017
vrpnc2	1.026	1.028	1.035	1.037	1.023
vrpnc12	1.003	1.003	1.002	1.0	1.001
vrpnc3	1.036	1.045	1.049	1.039	1.041
vrpnc11	1.006	1.004	1.005	1.005	1.004
vrpnc4	1.056	1.062	1.057	1.063	1.053
vrpnc5	1.074	1.077	1.077	1.076	1.063

Genetic Algorithm

Overview


Initialise Population
Evaluate
Selection
Crossover
Mutation
Termination Criterion ?
No
Yes
Solution Set

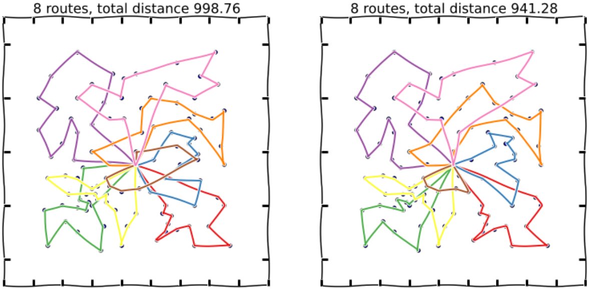
Crossover



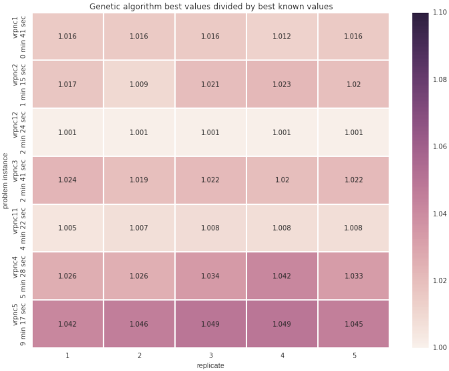
Mutation Description

- reuse the λ_1-interchange operator
- use a fixed (in this case 5) number of iterations
- serves as local exploration and changes clusters of clients

Mutation example



Results



Genetic algorithm best values divided by best known values

problem instance	1	2	3	4	5
vrpnc1 0 min 41 sec	1.016	1.016	1.016	1.012	1.016
vrpnc2 1 min 15 sec	1.017	1.009	1.021	1.023	1.02
vrpnc12 2 min 24 sec	1.001	1.001	1.001	1.001	1.001
vrpnc3 2 min 41 sec	1.024	1.019	1.022	1.02	1.022
vrpnc11 4 min 22 sec	1.005	1.007	1.008	1.008	1.008
vrpnc4 5 min 28 sec	1.026	1.026	1.034	1.042	1.033
vrpnc5 9 min 17 sec	1.042	1.046	1.049	1.049	1.045

Tabu Search

Tabu Search Description

- Neighbourhood Structure
- Tabu List
- Diversification

Neighbourhood Structure

- λ_1-interchange
- only feasible solutions

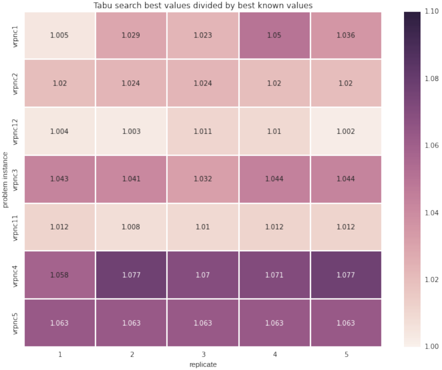
Tabu List

- avoid reversing a move
- remember pairs $(\mathit{client},\ \mathit{route})$
- $\mathit{max}\{7,\ -40 + 9.6 \times \mathit{ln}(n \times v)\}$

Diversification by Multi-Start

- takes a parameter called *patience*
- patience replenish after a new best is found
- patience runs out → random savings

Best Results in 60 sec



Tabu search best values divided by best known values

problem instance	1	2	3	4	5
vrpnc1	1.005	1.029	1.023	1.05	1.036
vrpnc2	1.02	1.024	1.024	1.02	1.02
vrpnc12	1.004	1.003	1.011	1.01	1.002
vrpnc3	1.043	1.041	1.032	1.044	1.044
vrpnc11	1.012	1.008	1.01	1.012	1.012
vrpnc4	1.058	1.077	1.07	1.071	1.077
vrpnc5	1.063	1.063	1.063	1.063	1.063

Overall Performance

Overall Performance

- complexity : [RS, GA, Tabu]
- speed: [Tabu, RS, GA]
- quality: [GA/Tabu, RS]

QA