



Published on *Oleksii Serdiuk* (<https://oleksii.name>)

[Home](#) > [Projects](#) > TSP Solver and Generator

TSP Solver and Generator ^[1]



Short description: Generate and solve Travelling Salesman Problems

<http://tspsg.info/>

Homepage:

Supported platforms:

- Windows
- Linux
- FreeBSD
- Windows Mobile
- Symbian
- BlackBerry 10

UI Languages:

- English
- Українська
- Русский

Technologies:

- C++
- Qt 4
- Qt 5

Description:

TSP Solver and Generator is intended to generate and solve **Travelling Salesman Problem** tasks. It uses *Branch and Bound* method for solving. Its input is a number of cities and a matrix of city-to-city travel costs. The matrix can be populated with random values in a given range (which is useful for generating tasks). The result is an optimal route, its price, step-by-step matrices of solving and a solving graph. The task can be saved in an internal binary format and opened later. The result can be printed or saved as *PDF*¹, *HTML*, or *ODF*.

TSPSG may be useful for teachers to generate test tasks or just for regular users to solve

TSP's. Also, it may be used as an example of using *Branch and Bound* method to solve a particular task.

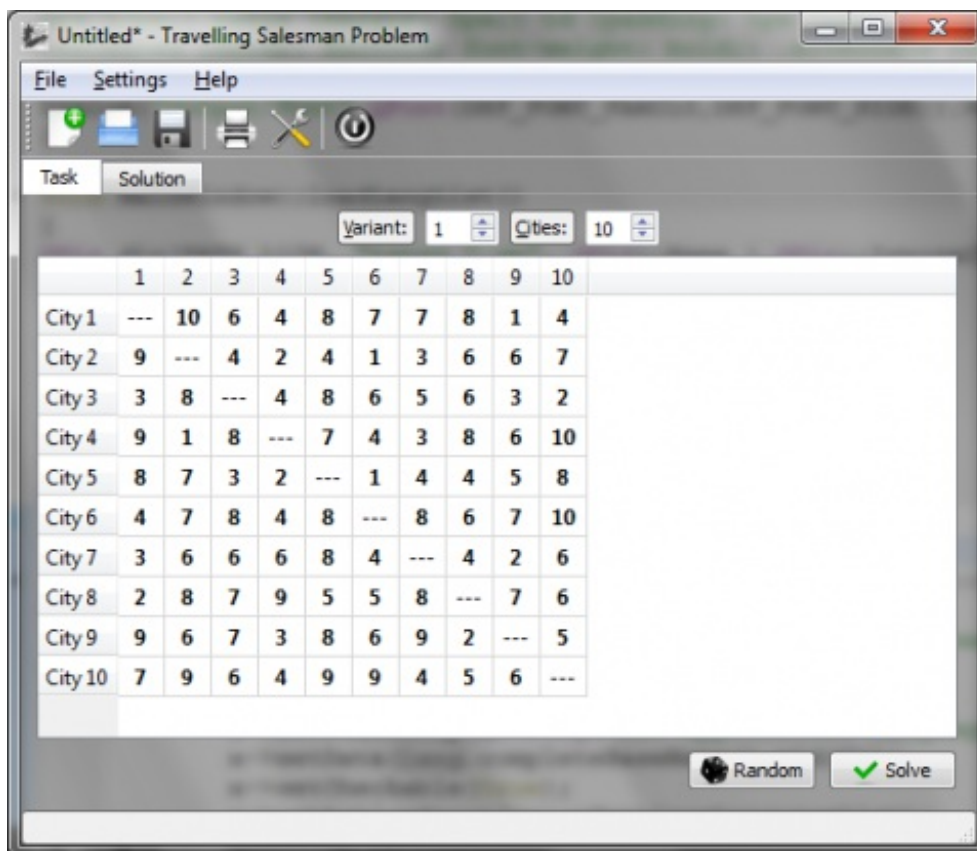
1. Depends on the platform. ↩

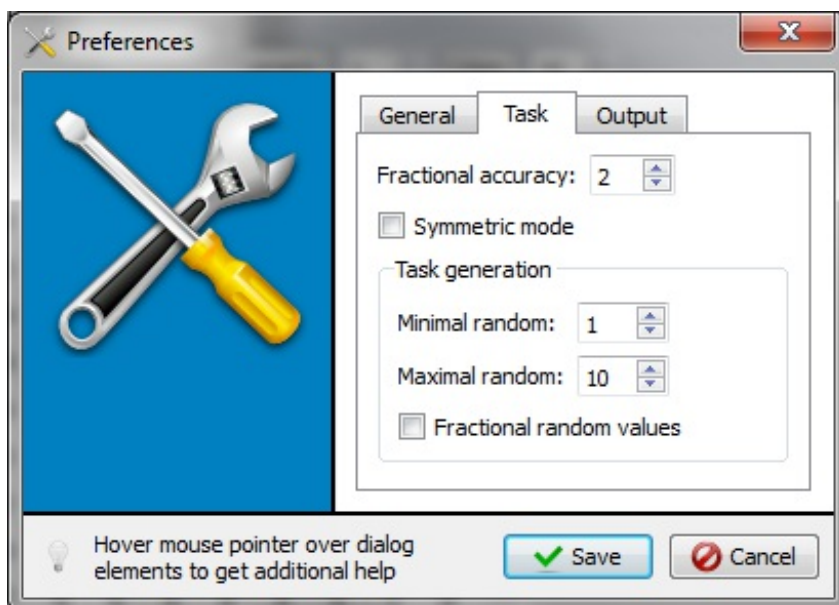
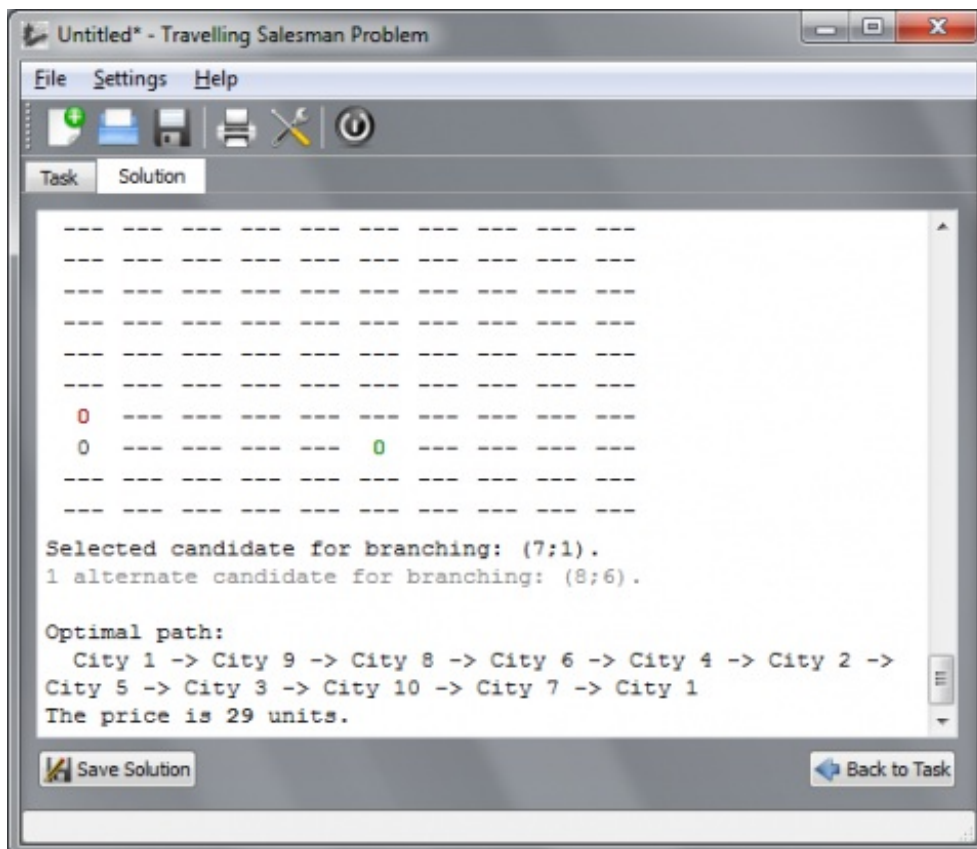
My contribution:

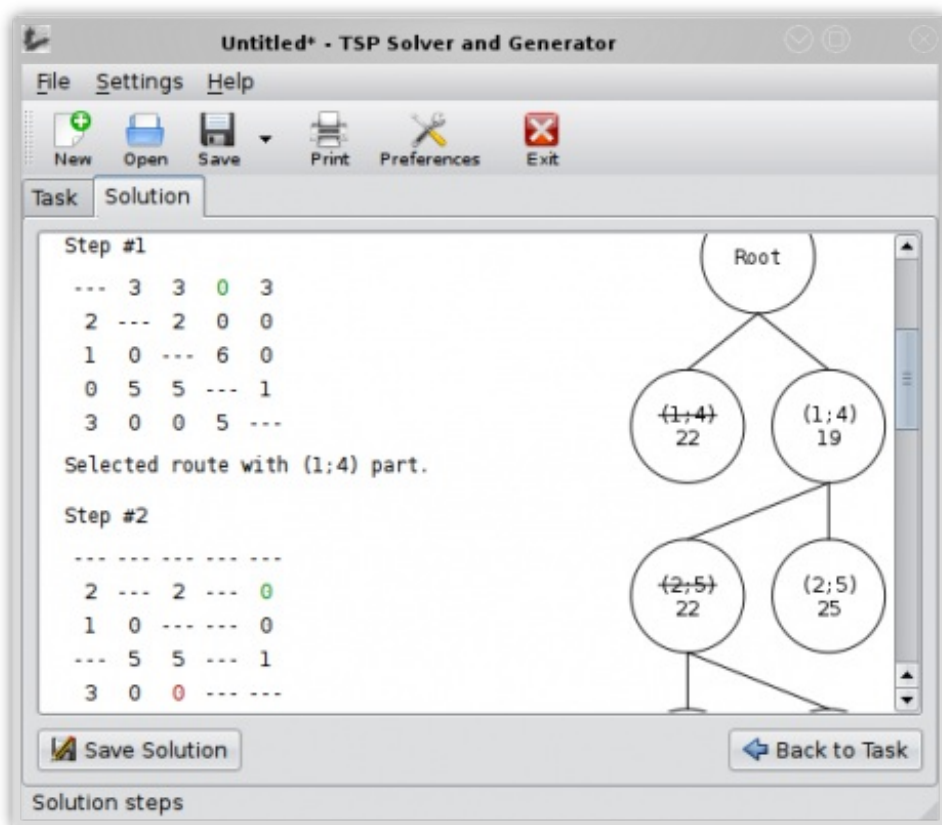
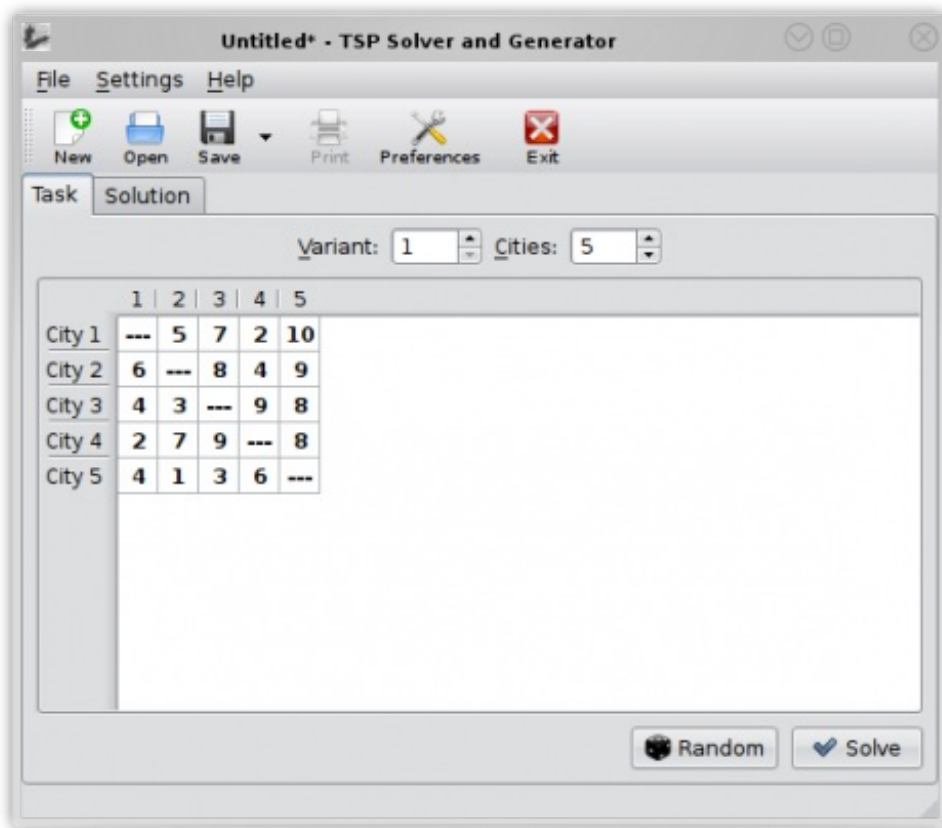
I am the author of this project. The project is currently “on hold” due to the lack of my spare time. But it will be revived, eventually.

This project was originally born as an Open Source rewrite of my University coursework [Branch and bound method of solving Travelling Salesman Problem](#)^[2].

Screenshots:







Solution Progress 19:32

Task Solution

Variant: 1 Cities: 5

	1	2	3	4	5
City 1	--	9	3	6	6
City 2					
City 3					
City 4					
City 5					

Random Solve

File Menu

Untitled* - TSP Solver and G 19:32

Task Solution

```

-----
--- 0 --- 3 ---
Selected route with (5;2)
part.

Step #4
-----
----- 0 -----
----- 1 0 -----
-----

Selected route with (2;4)
part.
1 alternate candidate for
branching: (3;5).

Optimal path:
City 1 -> City 3 -> City 5 -
-> City 2 -> City 4 -> City 1

```

Save Solution Back to Task

File Menu

File Settings Help

+ [Printer] [Disk] [Wrench] [Close]

Task Solution

Variant: 1 Cities: 5

	1	2	3	4	5
City 1	---	7	7	10	3
City 2	Generating solution output...				
City 3	[Cancel]				
City 4	3	6	7	---	2
City 5	9	3	3	4	---

[Random] [Solve]

File Settings Help

+ [Printer] [Disk] [Wrench] [Close]

Task Solution

Variant #1 Task

Task:

```

--- 7 7 10 3
3 --- 7 4 7
7 2 --- 10 6
3 6 7 --- 2
9 3 3 4 ---

```

Variant #1 Solution

Step #1

```

--- 4 4 6 0
0 --- 4 0 4
5 0 --- 7 4
1 4 5 --- 0
6 0 0 0 ---

```

Selected route with (1;5) part.
2 alternate

```

graph TD
    Root((Root)) --- N18((1;5) 18)
    Root --- N15((1;5) 15)
    N15 --- N15_1(( ))
    N15 --- N15_2(( ))

```

[Save Solution] [Back to Task]

Tags:

- Open Source
- Qt
- TSPSG

Short URL: <https://olse.me/HyJ>

Copyright © 2014-2018 **Oleksii Serdiuk**.

Source URL (modified on 05.06.2016 - 23:14): <https://oleksii.name/en/projects/tspsg>

Links

[1] <https://oleksii.name/en/projects/tspsg>

[2] <https://oleksii.name/en/projects/archived/zkommodrd>