Application of Nested Monte-Carlo methods to the Traveling Salesman Problem with Time Windows

Tristan Cazenave[1] and Fabien Teytaud[1,2]

[1] LAMSADE, Université Paris Dauphine

[2] HEC Paris, CNRS, 1 rue de la Libération 78351 Jouy-en-Josas

Outline

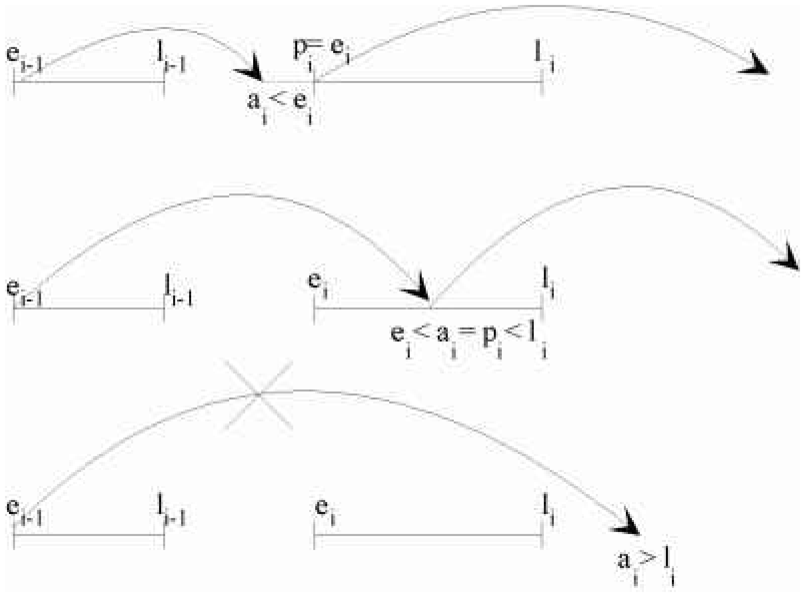
- Traveling Salesman Problem with Time Windows
- Nested Monte-Carlo Algorithm
- Nested Roll-out Policy Adaptation
- Experiments
- Conclusion

Traveling Salesman Problem (TSP)

- Data
  - List of cities
  - Distances between all cities
- Goal
  - Find a path visiting each city exactly once
  - The path must be as short as possible

Traveling Salesman Problem with Time Windows (TSPTW)

- Additionnal property: Time windows
  - A city can not be visited before a certain time and after a certain time



- Some problems have no solution
- Finding a valid solution is NP-hard

Outline

- Traveling Salesman Problem with Time Windows
- Nested Monte-Carlo Algorithm
- Nested Rollout Policy Adaptation
- Experiments
- Conclusion

Nested Monte-Carlo (NMC)

[Cazenave, 2009]

- Tree exploration algorithm
- Evaluation with Monte-Carlo simulations
- Particularly efficient for one player games and when late decisions are as important as early ones.

Nested Monte-Carlo (NMC)

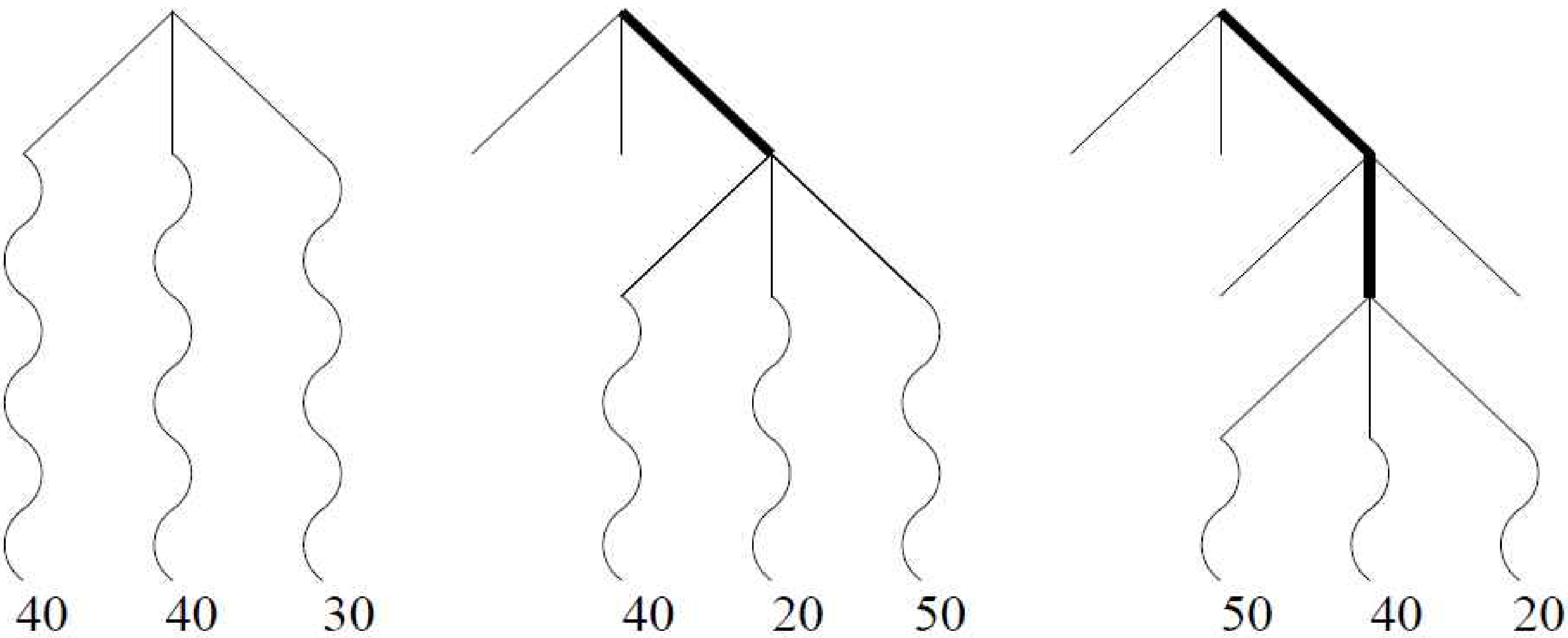
[Cazenave, 2009]

- Nested plays a whole game and returns the associated score
- Nested takes for parameters the level n and the current position (recursive algorithm)
- Principle
  - The score of an action is calculated by calling a nested with level n-1
  - The level 0 of NMC is a Monte Carlo simulation (random play until the end of the game)

NMC

- Level 0
  - Monte-Carlo policy
  - Choose a city randomly
- Level > 0
  - Launch NMC(*level*-1)
  - The action with the highest score is chosen

NMC(level=1) example



Adding Heuristics

[Rimmel et al, 2011]

- The algorithm can be improved by modifying the Monte Carlo simulations.
- Instead of uniformly random, the actions are chosen according to expert knowledge :
  - The distance to the last city
  - The waiting time (related to the inf bound of the time window)
  - The remaining time before the end of the time window

Outline

- Traveling Salesman Problem with Time Windows
- Nested Monte-Carlo Algorithm
- Nested Rollout Policy Adaptation
- Experiments
- Conclusion

Nested Rollout Policy Adaptation (NRPA)

- NMC can be improved by modifying the Monte Carlo simulations.

- Instead of random playouts, a policy is learned :
  - Increase the weights of the best cities
  - Decrease the weights of other cities
  - For each city : compute a probability proportional to the exp of its weight

[Rosin, 2011]

Nested Rollout Policy Adaptation (NRPA)

- Level 0
  - Adapted policy
  - Choose a city accordingly to its probability
- Level > 0
  - Do N iterations of NRPA(level -1)
  - Update
    - The scores
    - The sequences
    - The policy

Adding expert-knowledge (NRPA_EK)

- Force to visit cities as soon as they go after their windows end.
- Avoid visiting a city if it makes another city go after its windows end.
- Consider all moves if no move available after these two tests.
- Important point : Optimal moves can not be pruned with this expert knowledge

Outline

- Traveling Salesman Problem with Time Windows
- Nested Monte-Carlo Algorithm
- Nested Rollout Policy Adaptation
- Experiments
- Conclusion

Experiment protocol

- Experiments :
  - Tuning of NMC
  - Analyzes of N and the level (NRPA)
  - Comparison of NRPA and NRPA_EK on one problem.
  - Comparison of the best results found by NMC, NRPA and NRPA_EK on a set of standardized problems

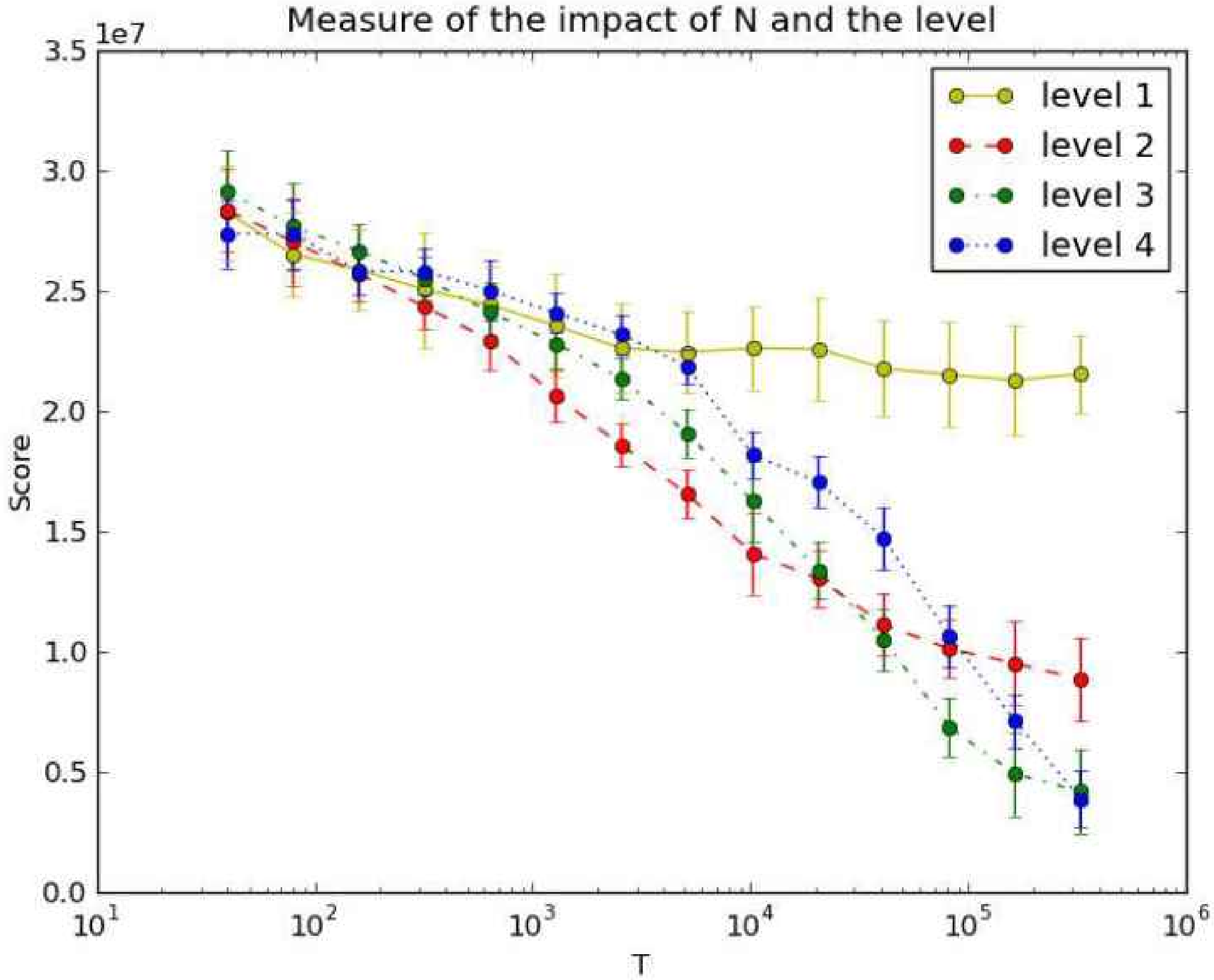
[Lopez-Ibanez and Blum, 2010]

Experiments (Tuning of NMC)

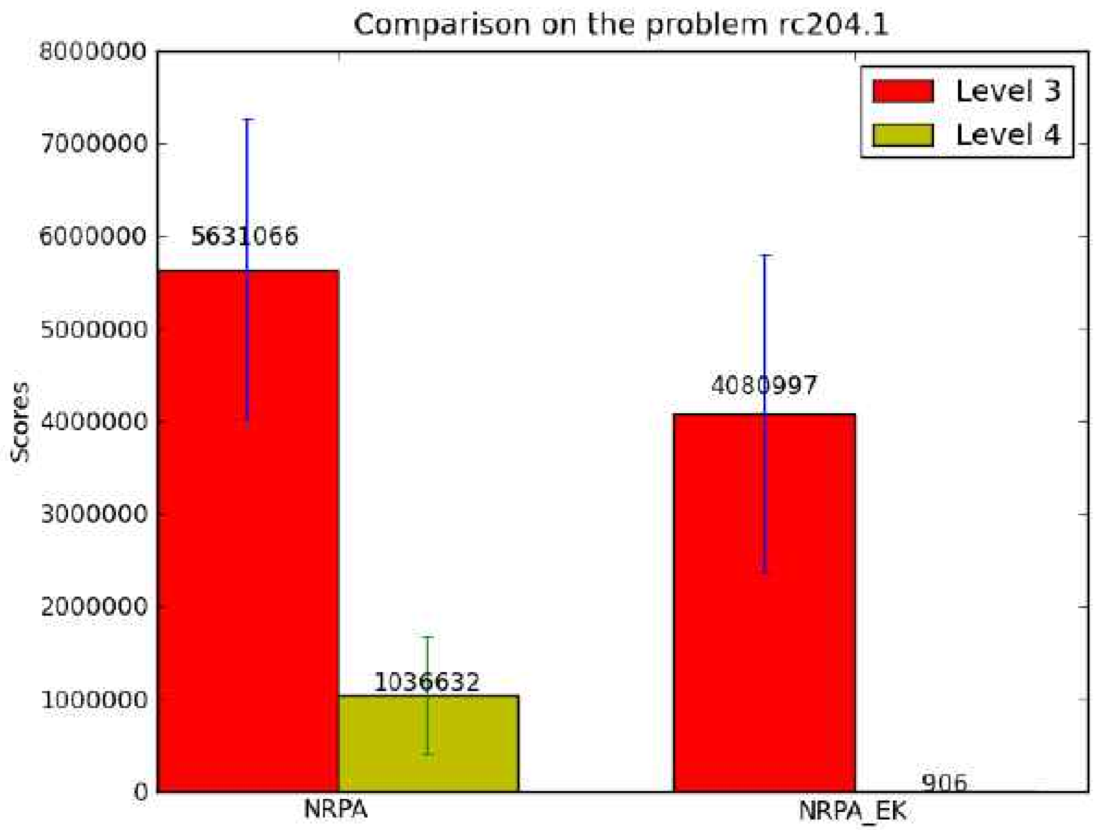
Iterations	BEST	KBEST	MEAN
1	2.7574e+06	2.4007e+06	2.3674e+06
2	5.7322e+04	3.8398e+05	1.9397e+05
3	7.2796e004	1.6397e+05	618.22
4	5.7274e+04	612.60	606.68
5	2.4393e+05	601.15	604.10
6	598.76	596.02	602.96
7	599.65	596.19	603.69
8	598.26	594.81	600.79
9	596.98	591.64	602.54
10	595.13	590.30	600.14
11	590.62	591.38	600.68
12	593.43	589.87	599.63
13	594.88	590.47	599.24
14	590.60	589.54	597.58
15	589.07	590.07	599.73

Table 1. Evolution of the true score on the problem rc206.3.

Experiment results (1)



Experiment results (2)



- Hardest problem from the set,
- 46 cities,
- Best known result : 868,76

Experiment results (3)

Problem	# cities	State of the art	NMC_EK score	NRPA score	NRPA_EK score
rc206.1	4	117.85	**117.85**	**117.85**	**117.85**
rc207.4	6	119.64	**119.64**	**119.64**	**119.64**
rc202.2	14	304.14	**304.14**	**304.14**	**304.14**
rc205.1	14	343.21	**343.21**	**343.21**	**343.21**
rc203.4	15	314.29	**314.29**	**314.29**	**314.29**
rc203.1	19	453.48	**453.48**	**453.48**	**453.48**
rc201.1	20	444.54	**444.54**	**444.54**	**444.54**
rc204.3	24	455.03	**455.03**	**455.03**	**455.03**
rc206.3	25	574.42	**574.42**	**574.42**	**574.42**
rc201.2	26	711.54	**711.54**	**711.54**	**711.54**
rc201.4	26	793.64	**793.64**	**793.64**	**793.64**
rc205.2	27	755.93	**755.93**	**755.93**	**755.93**
rc202.4	28	793.03	**793.03**	800.18	**793.03**
rc205.4	28	760.47	**760.47**	765.38	**760.47**

Experiment results (3)

Problem	# cities	State of the art	NMC_EK score	NRPA score	NRPA_EK score
rc202.3	29	837.72	**837.72**	839.58	839.58
rc208.2	29	533.78	536.04	537.74	**533.78**
rc207.2	31	701.25	707.74	702.17	**701.25**
rc201.3	32	790.61	**790.61**	796.98	**790.61**
rc204.2	33	662.16	675.33	673.89	664.38
rc202.1	33	771.78	776.47	775.59	772.18
rc203.2	33	784.16	**784.16**	**784.16**	**784.16**
rc207.3	33	682.40	687.58	688.50	**682.40**
rc207.1	34	732.68	743.29	743.72	738.74
rc205.3	35	825.06	828.27	828.36	**825.06**
rc208.3	36	634.44	641.17	656.40	650.49
rc203.3	37	817.53	837.72	820.93	**817.53**
rc206.2	37	828.06	839.18	829.07	**828.06**
rc206.4	38	831.67	859.07	831.72	**831.67**
rc208.1	38	789.25	797.89	799.24	793.60
rc204.1	46	868.76	899.79	883.85	880.89

Outline

- Traveling Salesman Problem with Time Windows
- Nested Monte-Carlo Algorithm
- Nested Rollout Policy Adaptation
- Experiments
- Conclusion

Conclusion

- Results
  - Efficient algorithm (77% of SOTA scores for NRPA_EK)
  - Promising results with no/few domain knowledge.
  - Expert knowledge is always helpful
  - Difficulties when the number of nodes becomes too large.

- Current work
  - Beam NRPA
  - Local optima issues ?

Thank you